

Figure : Typical partition between firmware and hardware

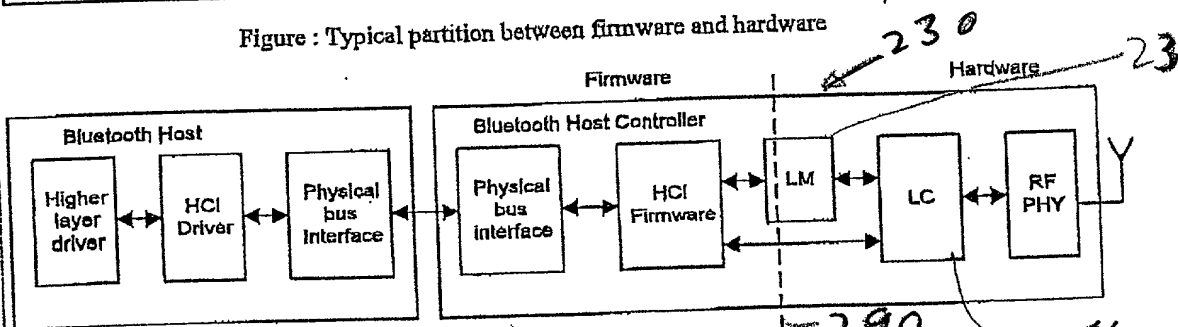


Figure : KC Technology's Partition between firmware and hardware

CONFIDENTIAL

FIG. 2
PRIOR
ART

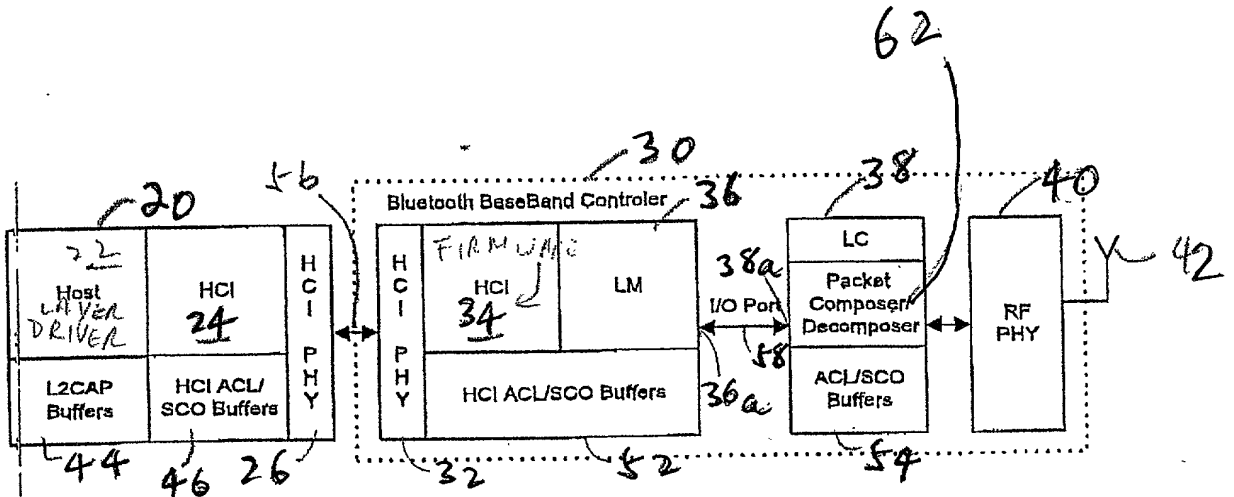


FIG. 3A
PRIOR
ART

Figure : HCI ACL Data Packet

31	24	23	16	15	8	7	0
Data Total Length			BC	PB	Connection Handle		
...			Data byte 1		Data byte 0		

Figure : HCI SCO Data Packet

31	24 23	16 15	8 7	0
Data byte 0	Data Total Length	Reserved	Connection Handle	
		...	Data byte 1	

FIG. 3B
PRIOR
ART

FIG. 4
PRIOR
ART

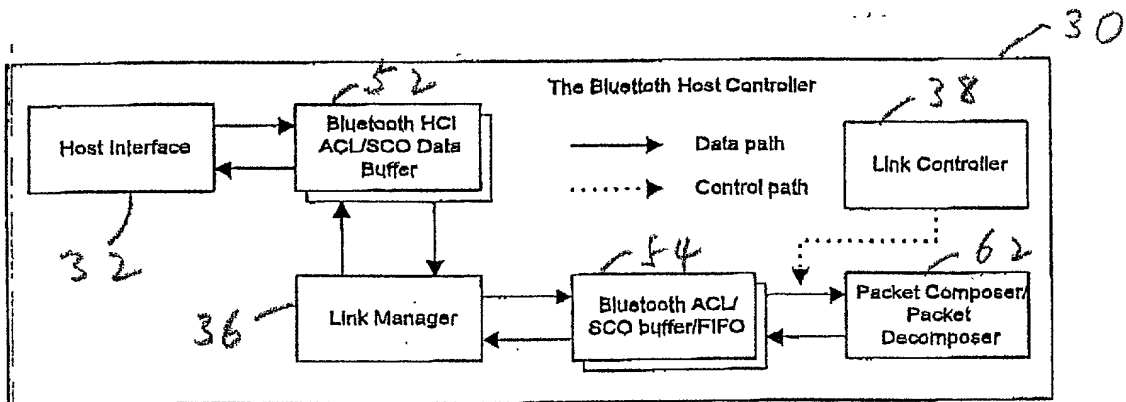


Figure : Data flow and Buffer scheme in the Bluetooth Host Controller

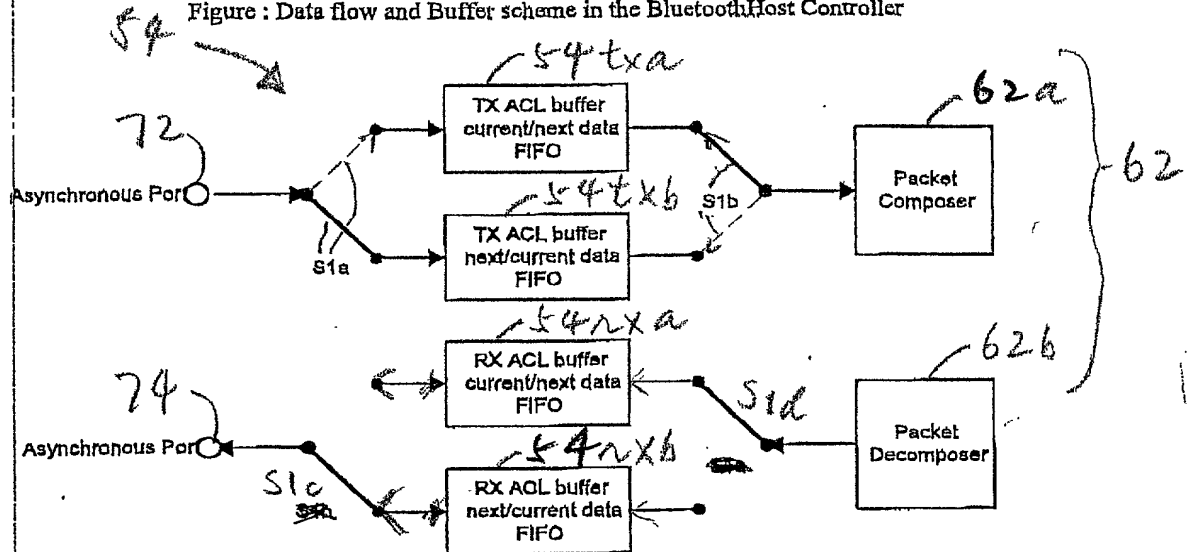


Figure : Dual buffer scheme for ACL packet transmission

10062877, 100601

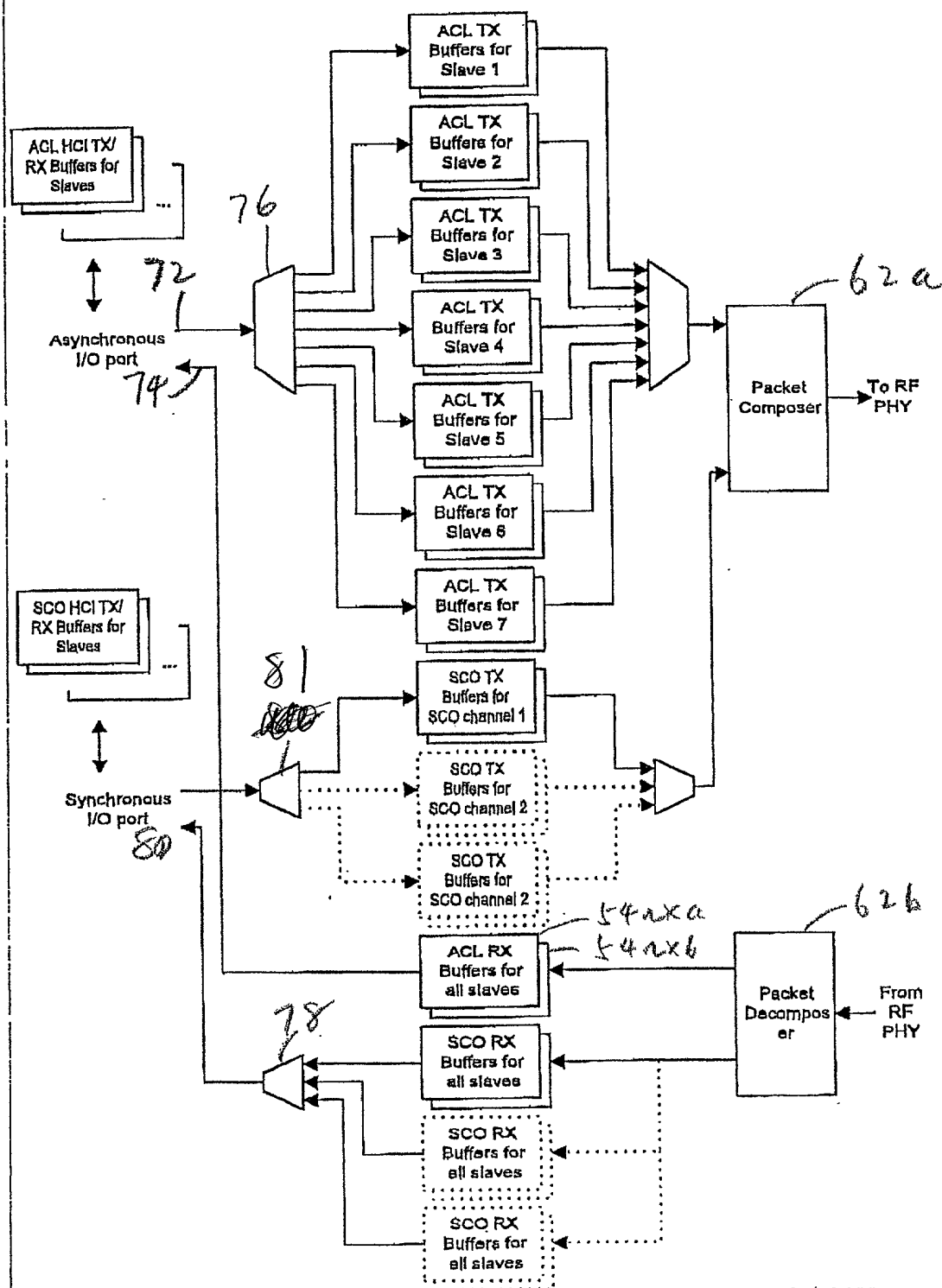
FIG. 5A
PRIOR
ART

CONFIDENTIAL

KC Technology Inc.

8/2/00

Confidential Information



CONFIDENTIAL

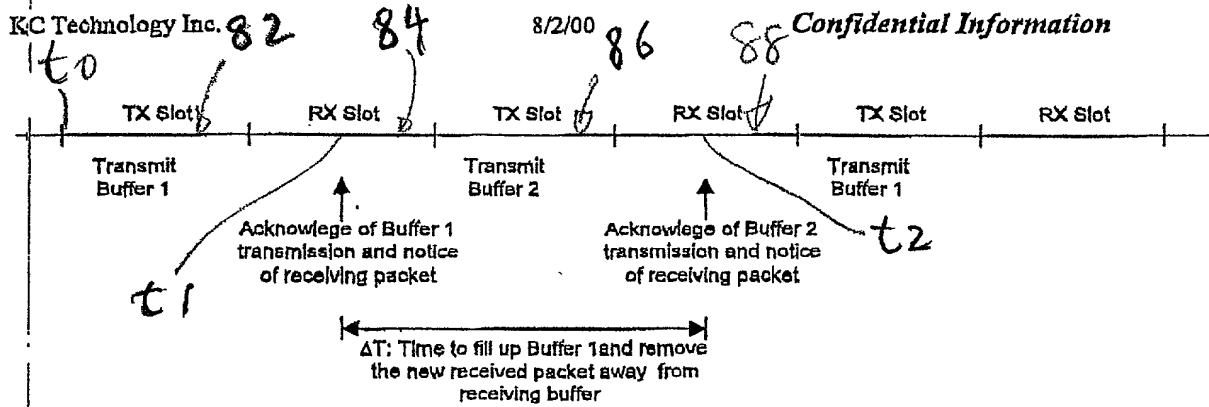


Figure : The worst case timing (ΔT) for the LM to load a TX buffer and unload a RX buffer

FIG.6
PRIOR
ART

CONFIDENTIAL

10053377 102601

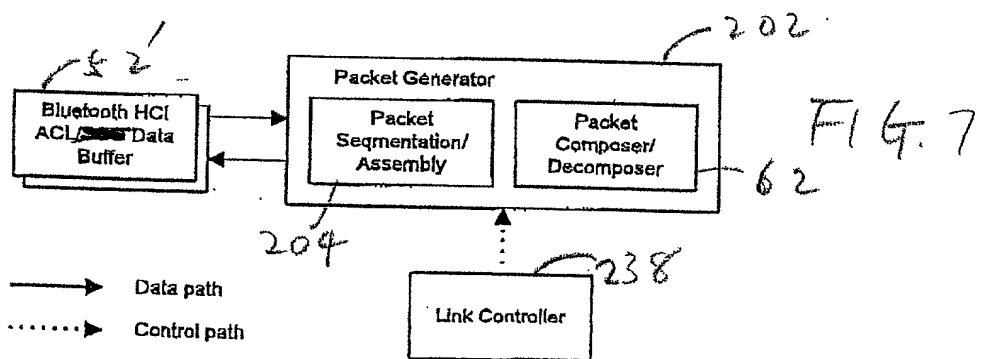


Figure : The Packet Generator accesses the HCI ACL/SCO buffers directly

10063377 100601

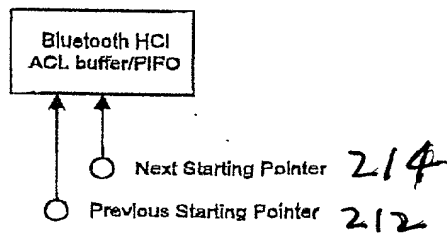
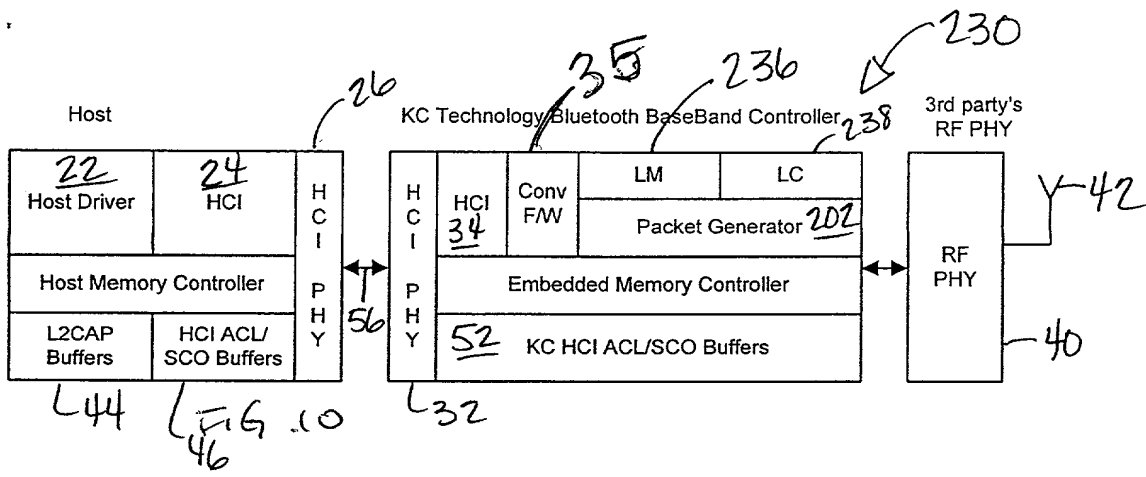


FIG. 8



KC Technology Inc.

8/2/00

Confidential Information

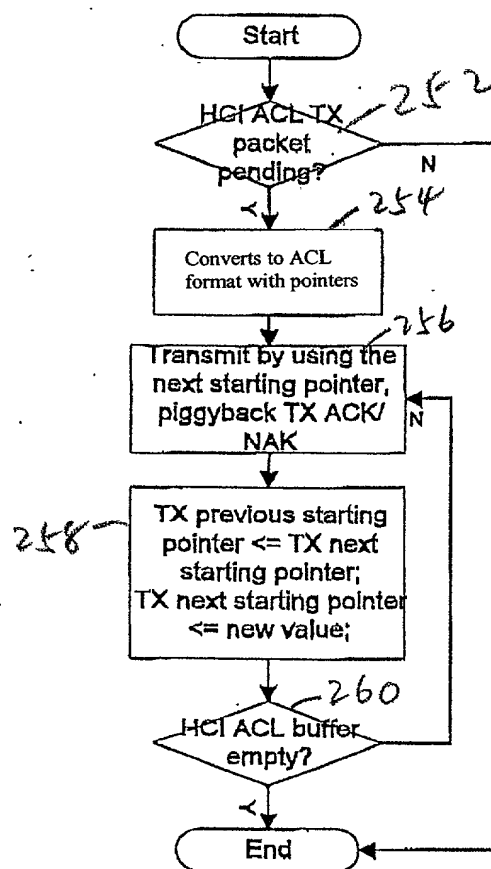


Figure : ACL transmission control flow of KC Technology's partition

FIG. 11

CONFIDENTIAL

KC Technology Inc.

8/2/00

Confidential Information

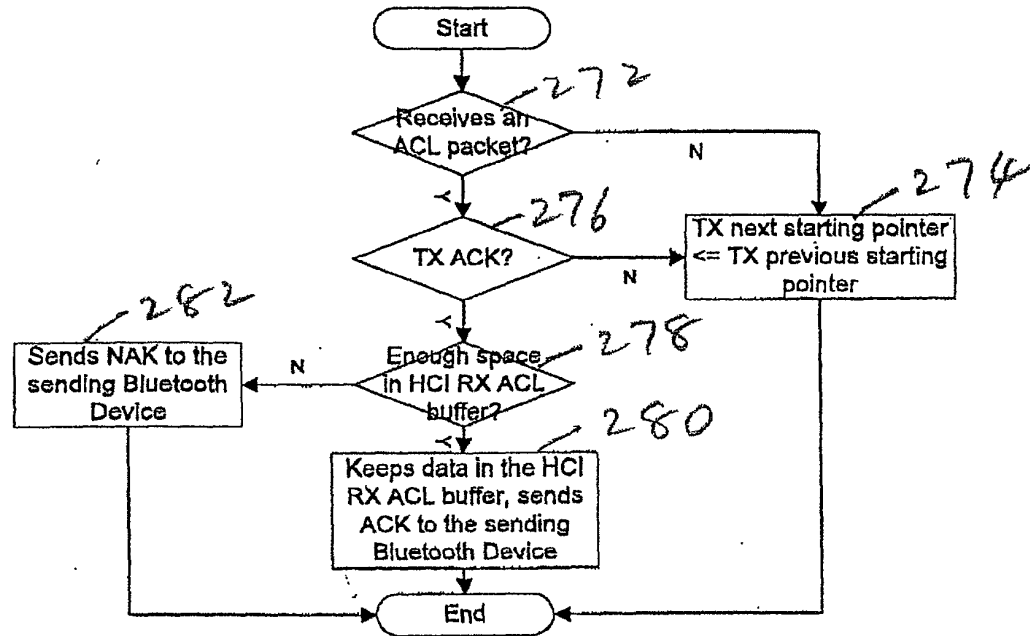


Figure : ACL receiving control flow of KC Technology's partition

FIG. 12

FIG. 14A

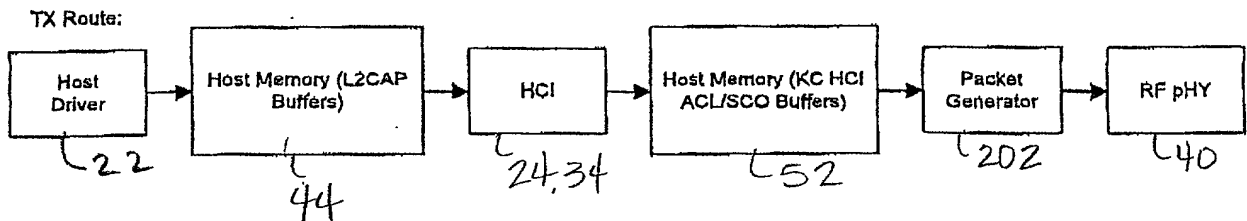
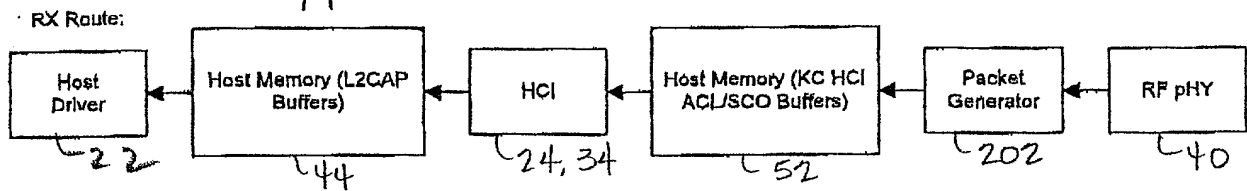


FIG. 14B



Type: A

FIG. 15C

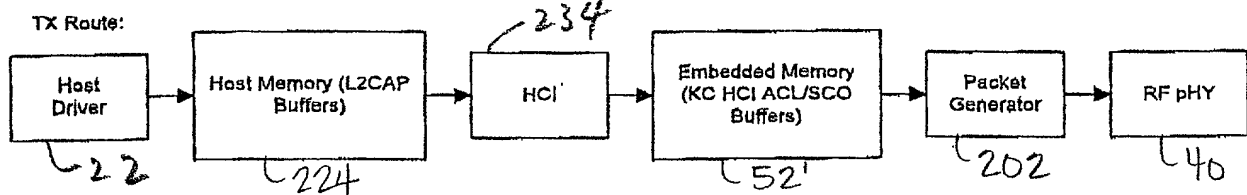
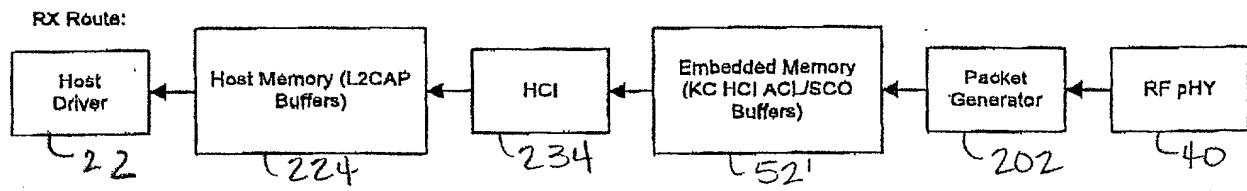


FIG. 15D



Type: B

FIG. 17C

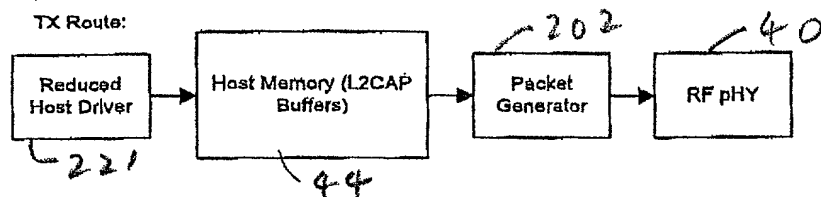
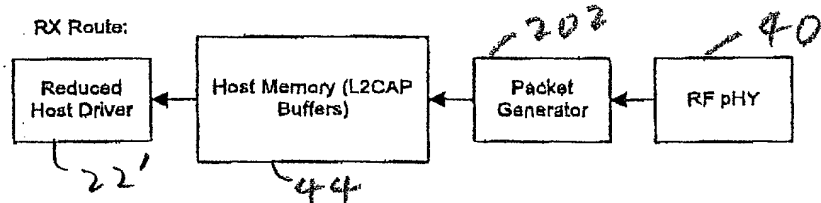


FIG. 17D



Type: A-2

Host / KC Technology Bluetooth BaseBand Controller

3rd party's
RF PHY

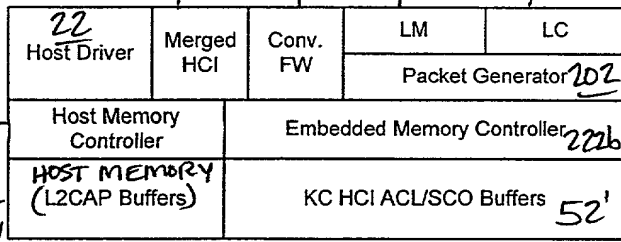


FIG. 15A

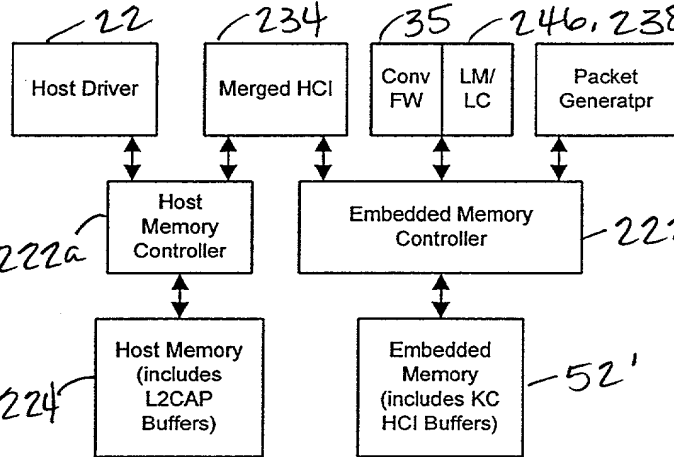


FIG. 15B

FIG. 15A

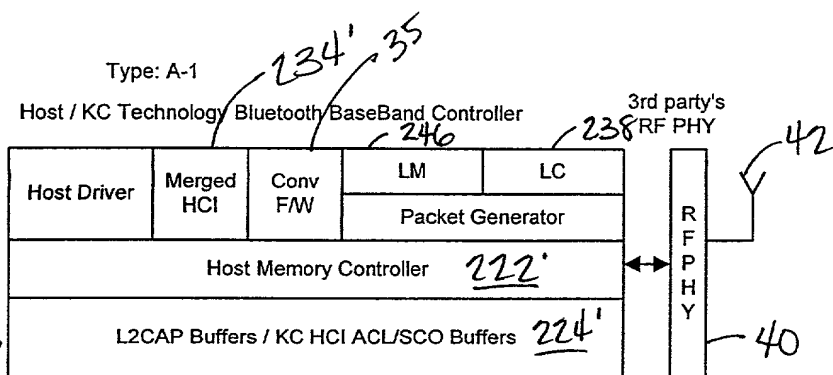


FIG. 16 A

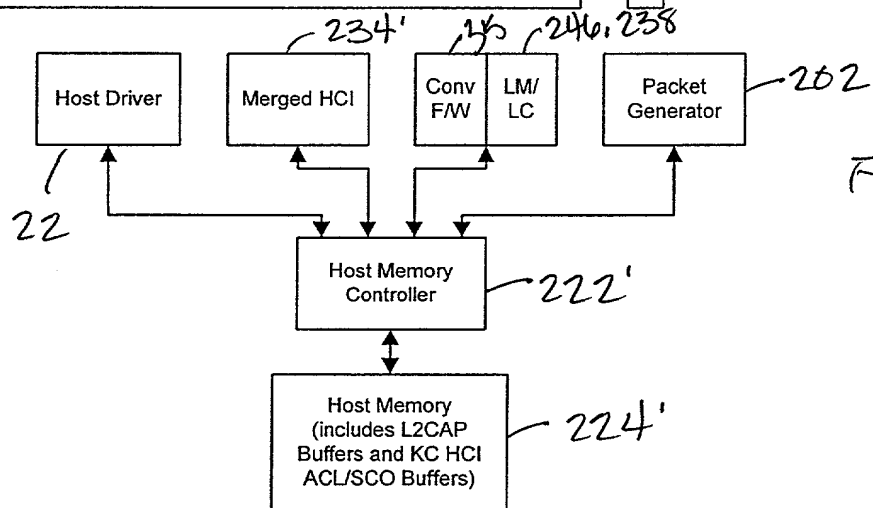
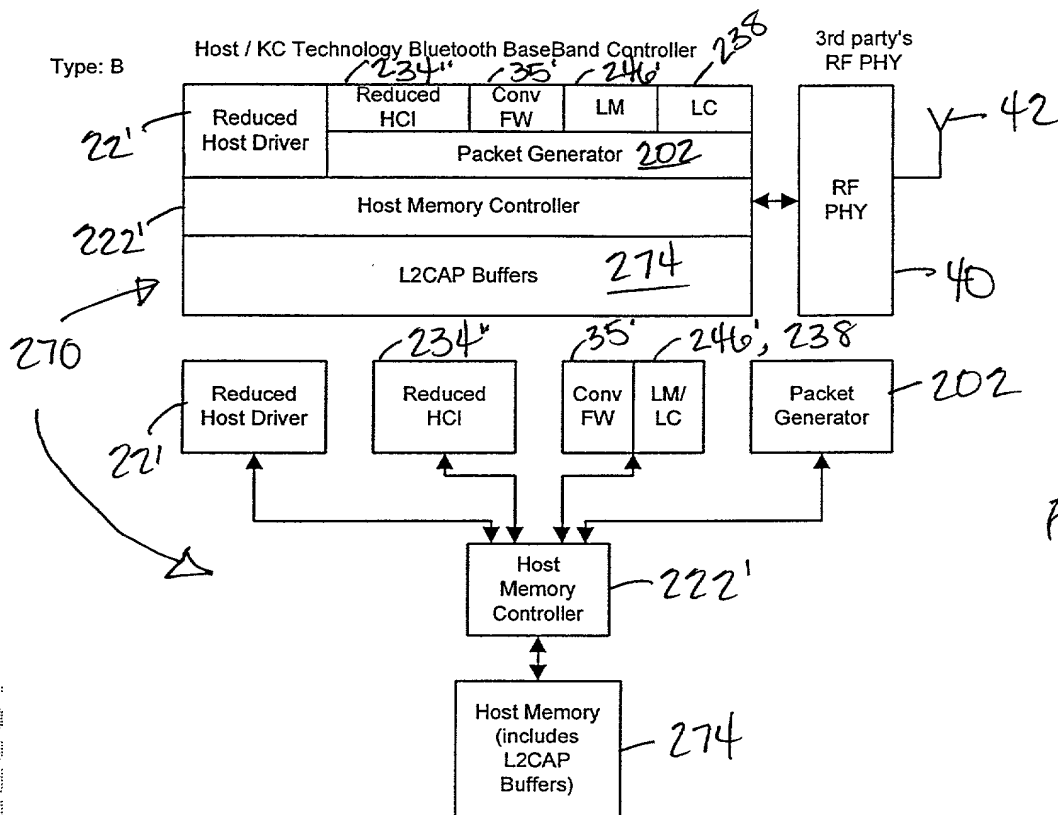


FIG. 16 B



400628277 400604

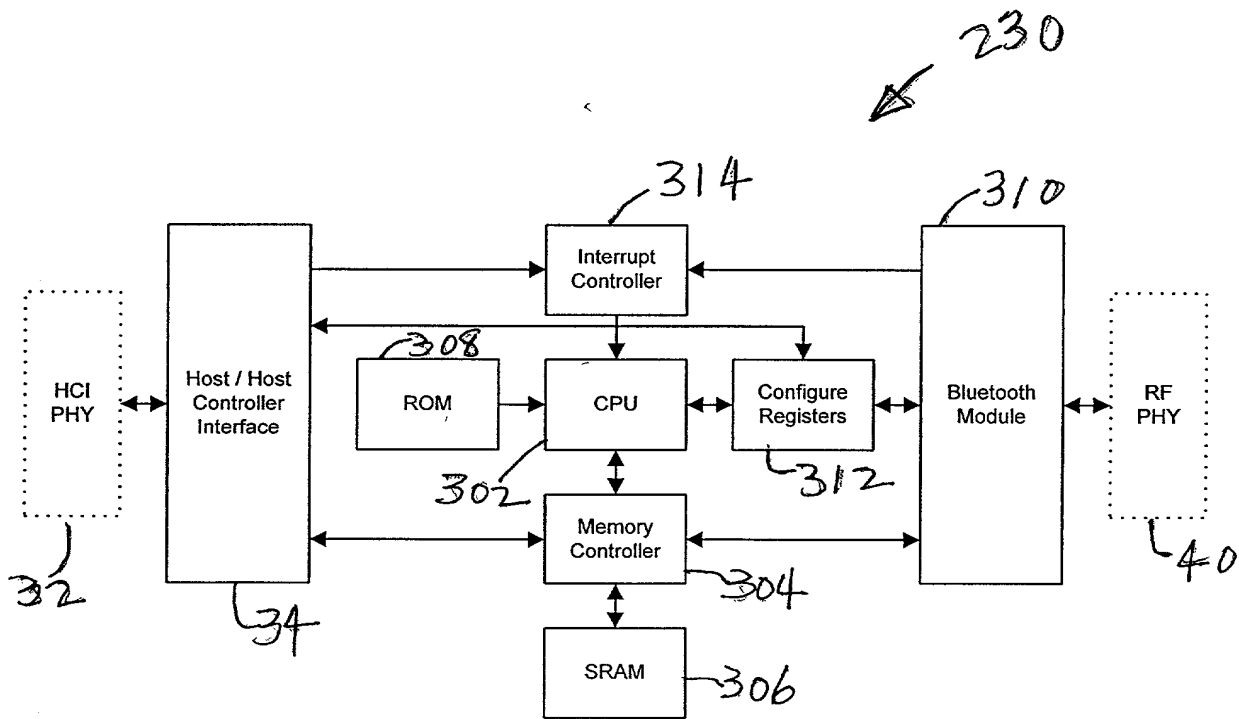
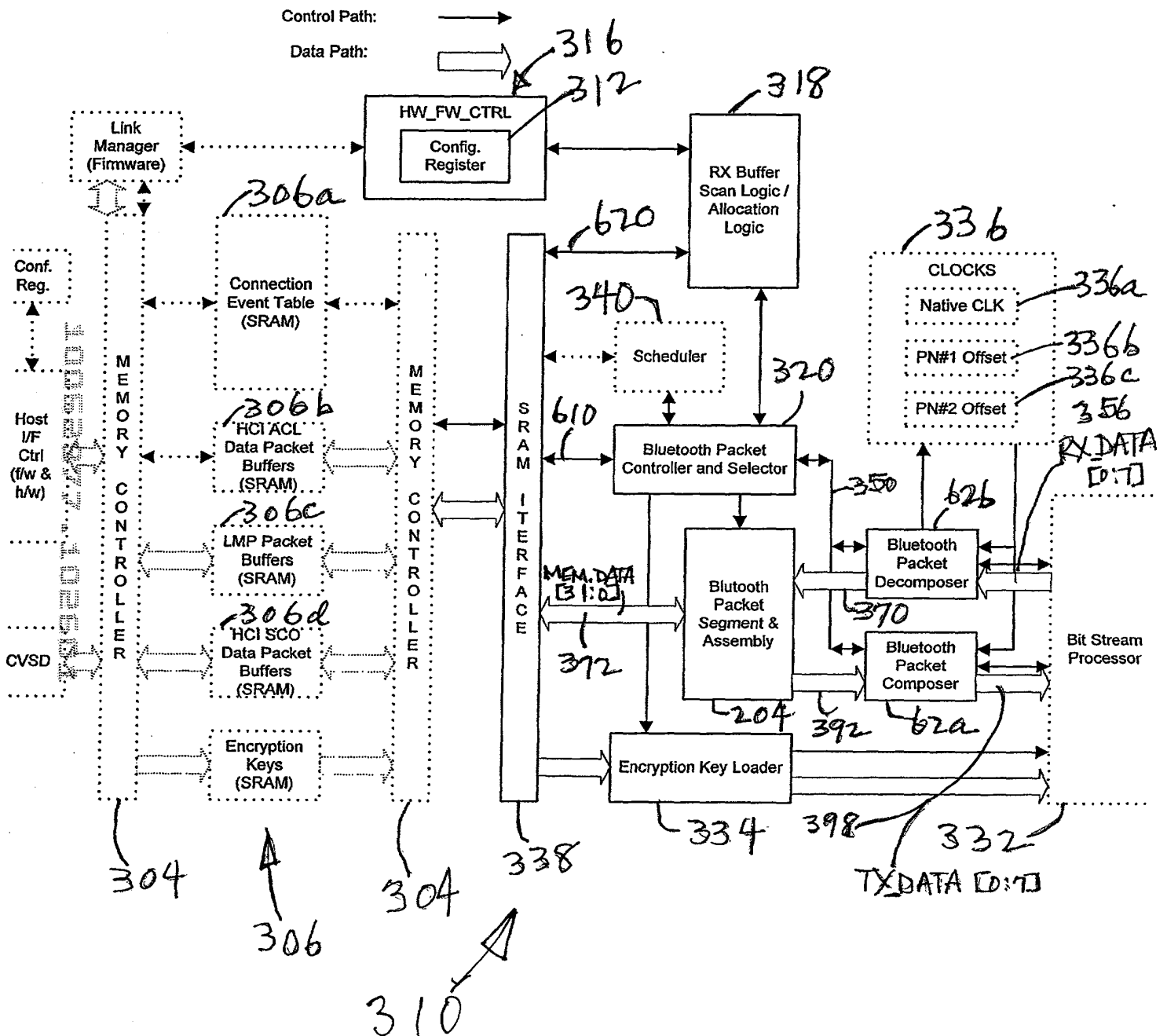


FIG. 18

9/1001 doc

Anything drawn in dashed line is not located inside this module



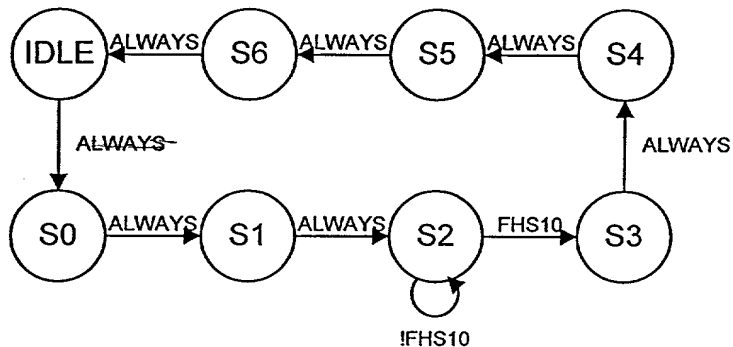
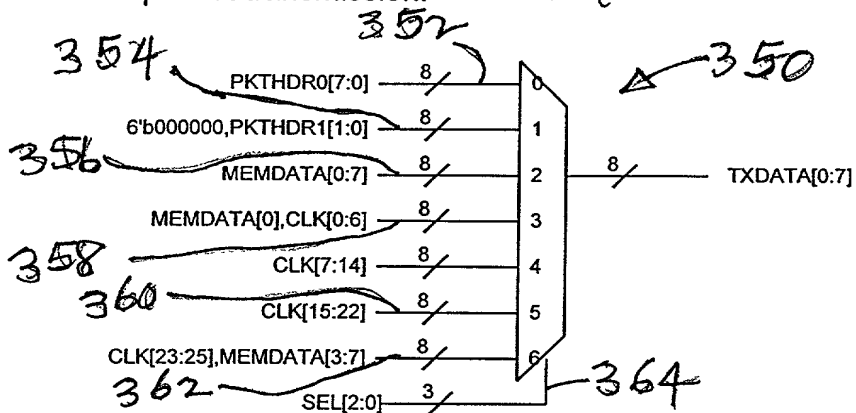


FIG. 24B

State	SEL[2:0]
IDLE	0
S0	0
S1	1
S2	2
S3	3
S4	4
S5	5
S6	6

Incoming FHS Packet storage format:

31				24 23				16 15				8 7				0			
na				0 1 1 1 0				BC 0 0 1				na							
UAP [1:0]		SP[1:0]		SR[1:0]		na		LAP											
CLASS[9:0]						NAP[15:0]						UAP[7:2]							
CLK[16:2]								AM_ADR [2:0]		CLASS[23:10]									
na	CLKOFFSET[16:2]								na		PSM[2:0]		CLK[27:17]						

Outgoing FHS Packet storage format:

31										24 23				16 15				8 7				0	
na										0	1	1	1	0	BC		0	0	na				
UAP [1:0]		SP[1:0]		SR[1:0]		na		LAP															
CLASS[9:0]						NAP[15:0]										UAP[7:2]							
na										AM_ADR [2:0]		CLASS[23:10]											
na										na				PSM[2:0]		na							

KC Technology, Confidential Information

FIG. 20

[illegible]

FIG. 22

DM1 packet transmission:

Page 1 of 1

TXDATA[7:0], RXDATA[7:0]: please refer to the definition on page 2 of FHS packet transmission.

Packet Generator, Bitstream Process Module diagram: please refer to the diagram on page 2 of FHS packet transmission.

Data Byte Sending Sequence in DM1 packet:

FIG. 25

Byte# & name	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
1:Packet Header 0	amadr0	amadr2	amadr1	pkttp0	pkttp1	pkttp2	pkttp3	flow
2:Packet Header 1	arqn	seqn	-	-	-	-	-	-
3:Payload Header 0	Lch 0	Lch 1	pld_flow	length0	length1	length2	length3	length4
4:ACL data 1	data bit 0	data bit 1	data bit 2	data bit 3	data bit 4	data bit 5	data bit 6	data bit 7
5: ACL data 2	data bit 0	data bit 1	data bit 2	data bit 3	data bit 4	data bit 5	data bit 6	data bit 7
...
n+3:ACL data n	data bit 0	data bit 1	data bit 2	data bit 3	data bit 4	data bit 5	data bit 6	data bit 7

n: data length

10062227 102504

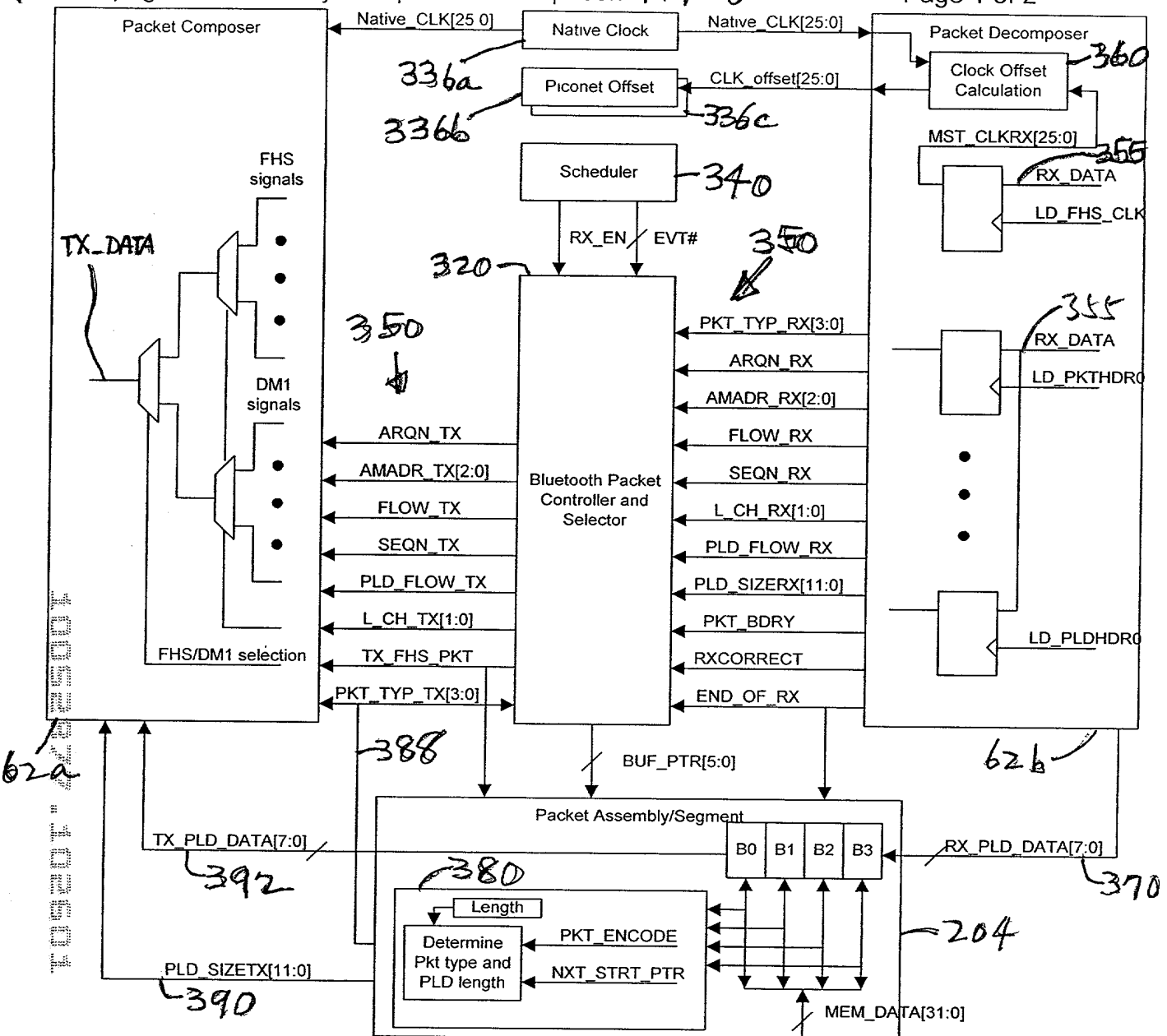


Figure 3: ACL Buffer while Sending

31	24	23	16	15	8	7	0
Flow	na	Data Total Length			BC	PB	Fv
							Flush expiration Time[12:2]
							Data byte 0
flsh	na	Next Starting Pointer			Pkt encode	Previous Starting Pointer	

Pkt_encode: In normal mode, this coding is listed as below. In One_Pkt_Mode, it represents the packet type code that is defined in the Bluetooth specification.

Pkt_encode[3:2]: 00 – AUX1 only; 01 – DM only; 10 – DH only; 11 – Automatic best fit

Pkt_encode[1:0]: 00 – Single slot packet; 01 – 3-slot packet; 10 – 5-slot packet; 11 – Reserved

Figure 5: ACL Buffer after Receiving in normal mode

31	24	23	16	15	8	7	0
Flow	na	Data Total Length			BC	PB	na
							Buffer releasing expiration time
							Data byte 0
na	Next Starting Pointer			Na	Previous Starting Pointer		

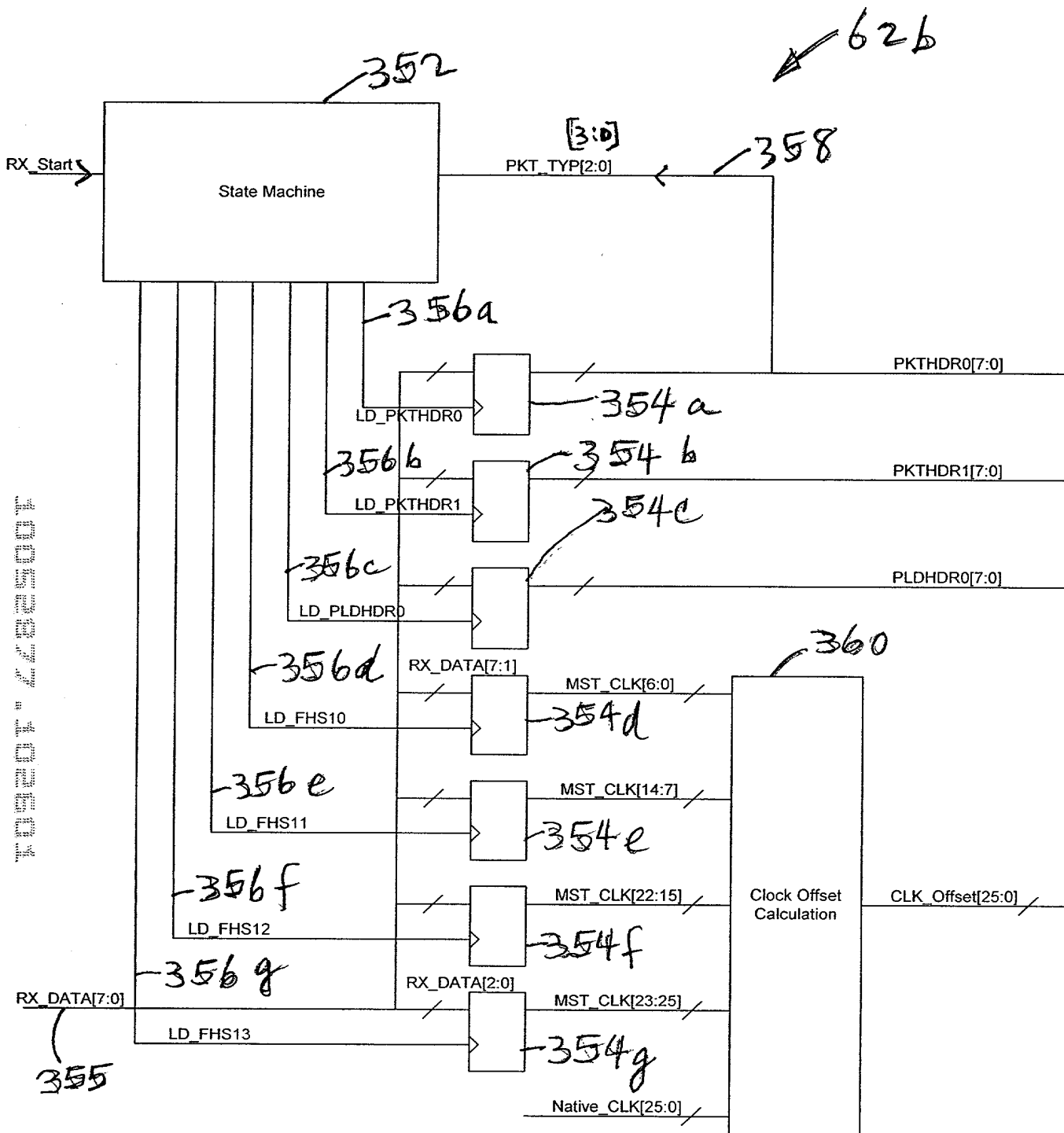


FIG. 27

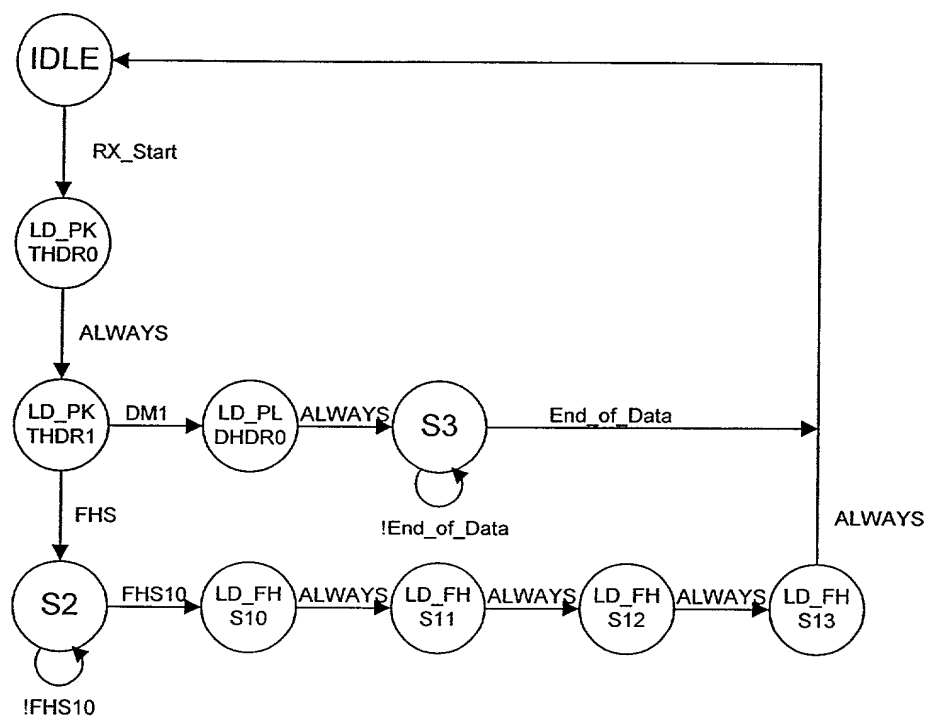


FIG. 28

Determine Pkt type and PLD length

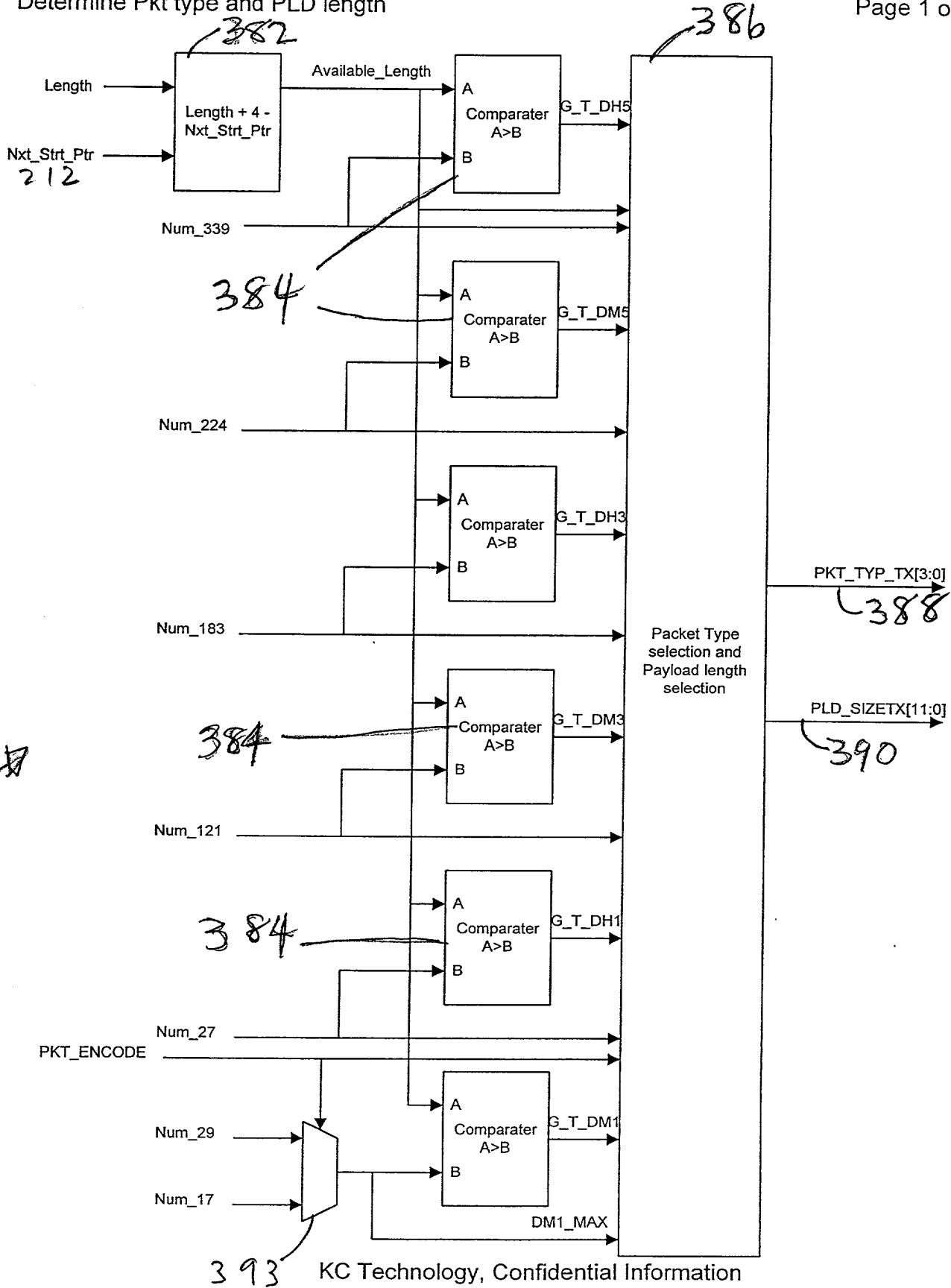
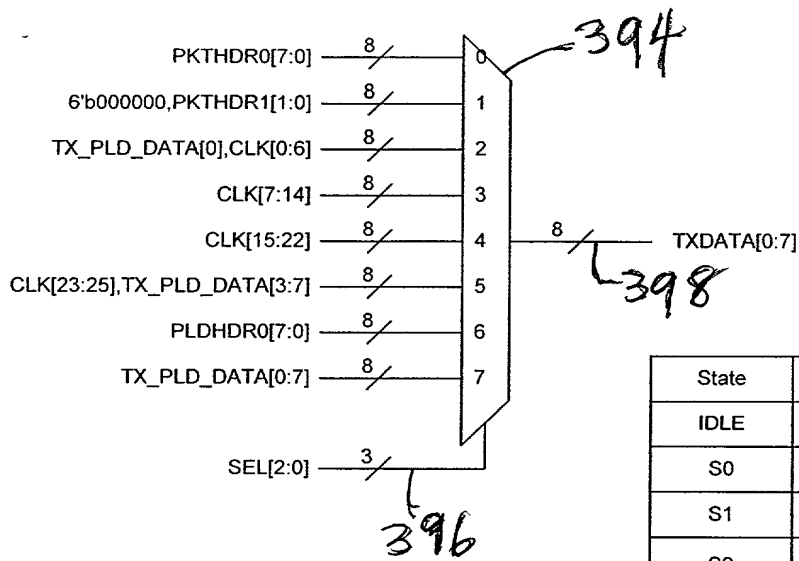


FIG. 30



State	SEL[2:0]
IDLE	0
S0	0
S1	1
S2	7
S3	2
S4	3
S5	4
S6	5
S7	6
S8	7

FIG. 31A

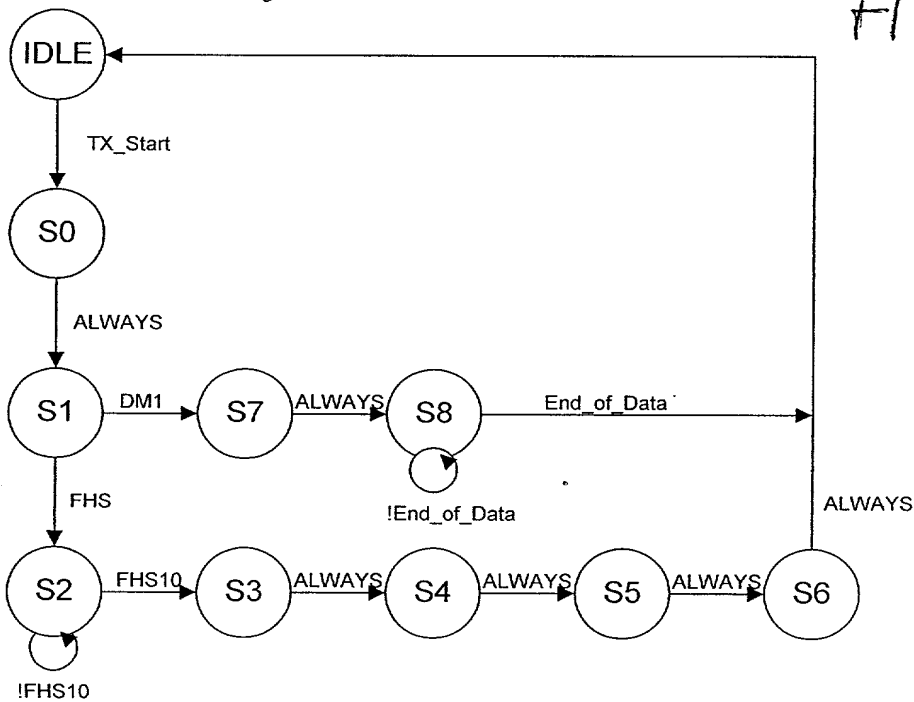
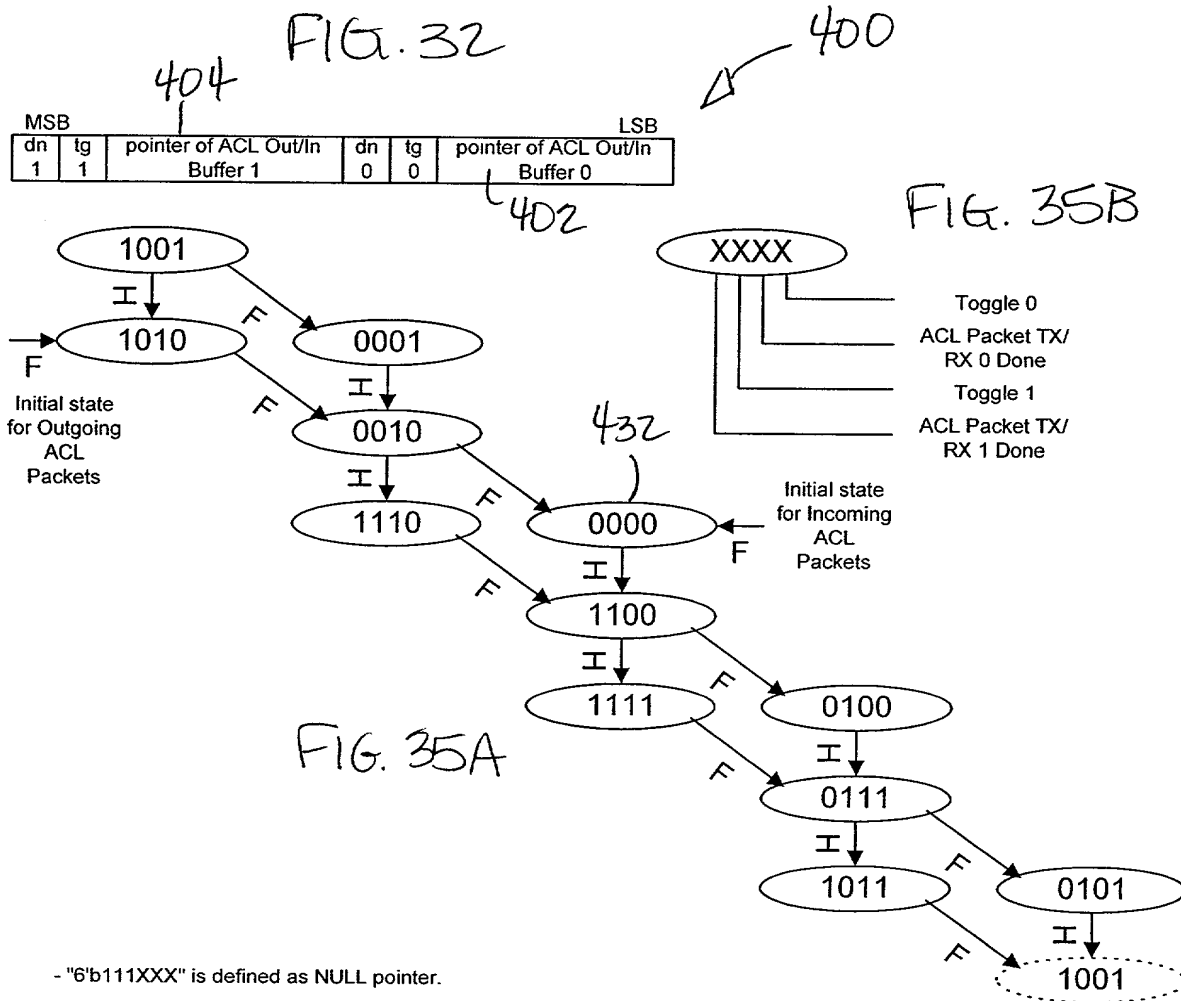


FIG. 31B

Dual pointer buffer control scheme (1)



- "6'b111XXX" is defined as NULL pointer.
- When Done bit is 1, firmware can update the pointer byte and hardware can only read this byte.
- When Done bit is 0, hardware can update the pointer byte and firmware can only read this byte.
- Toggle bit is changed by the hardware, it is toggled everytime when hardware finishes a task.
- This control scheme applies to the LMP out buffers as well.

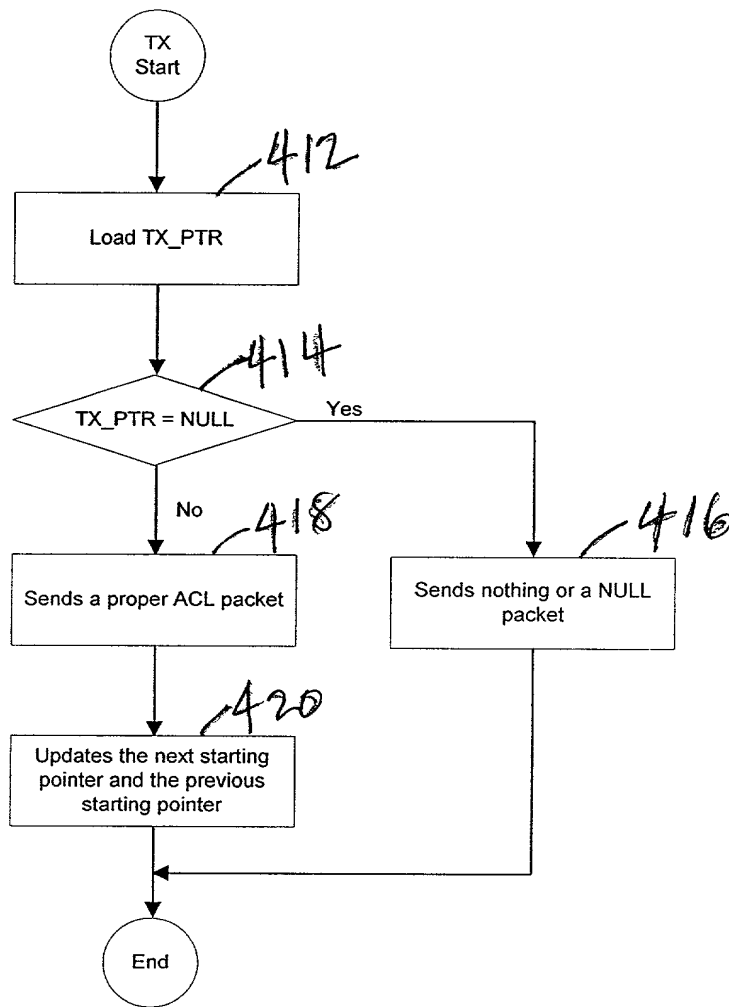


FIG. 33

FIG. 34A

L2CAP Packet Format																																																															
31								24								23								16								15								8								7								0							
Channel ID																L2CAP Length																																															
																...																Data byte 0																															
Data byte n																...																																															

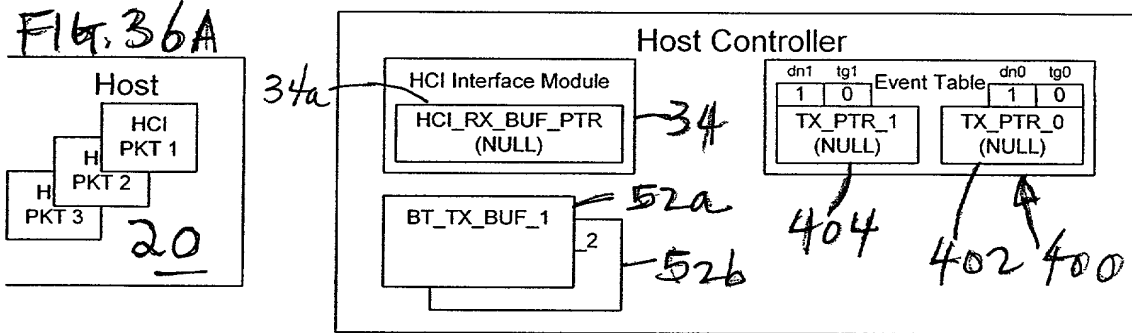
ACL Buffer while Sending																					
31		24				23		16				15		8				7		0	
Flow	na	Data Total Length								BC	1	0	Fv	Flush expiration Time[12:2]							
Channel ID								L2CAP Length													
								...								Data byte 0					
Data byte n								...													
flsh	na	4						Pkt encode				4									

2121

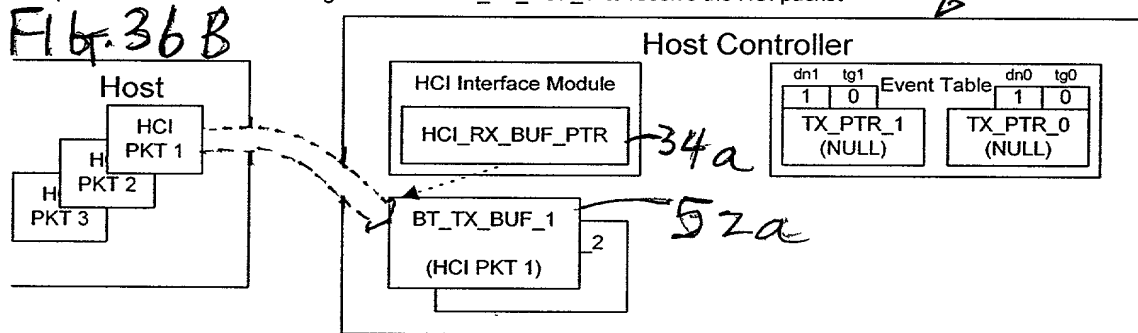
FIG. 34B

2141

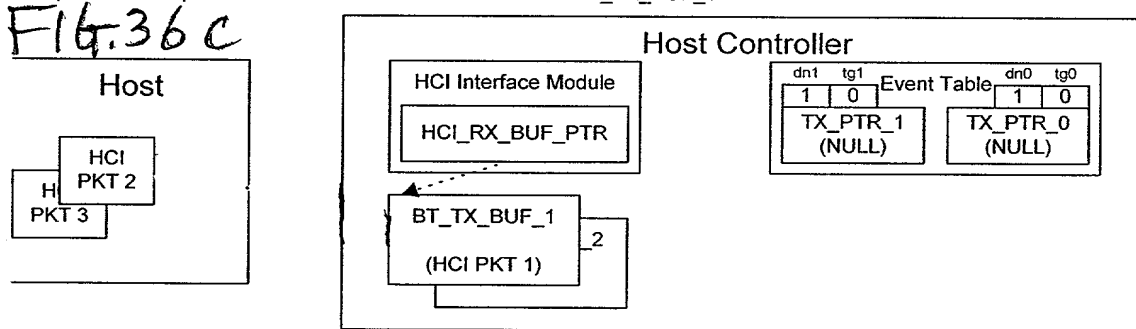
Step 1: After initialization, the value of pointers is 'NULL'



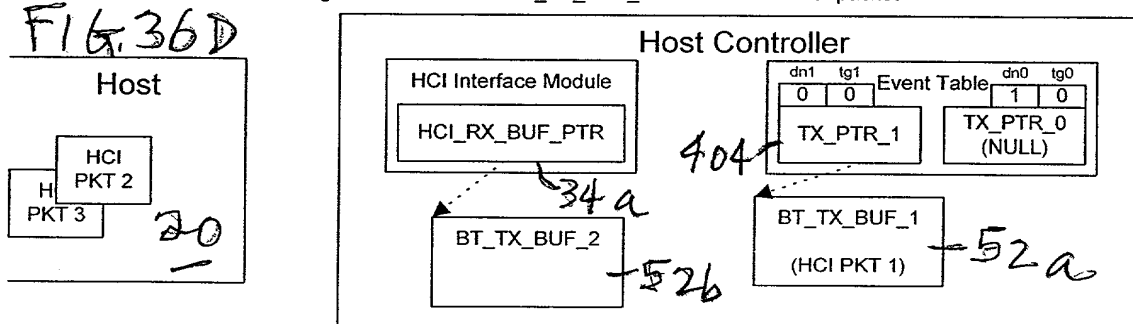
Step 2: The Host Controller assigns a buffer 'BT_TX_BUF_1' to receive the HCI packet



Step 3: The HCI packet 'HCI PKT 1' is stored in the buffer 'BT_TX_BUF_1'

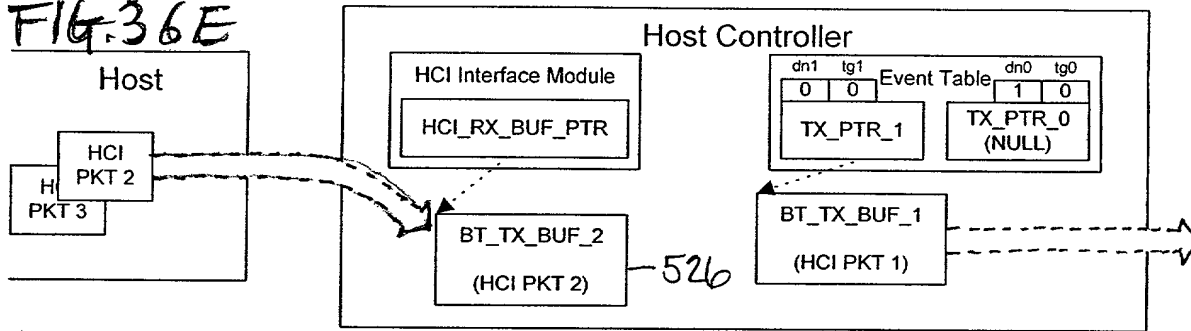


Step 4: The Host Controller assigns another buffer 'BT_TX_BUF_2' to receive the HCI packet



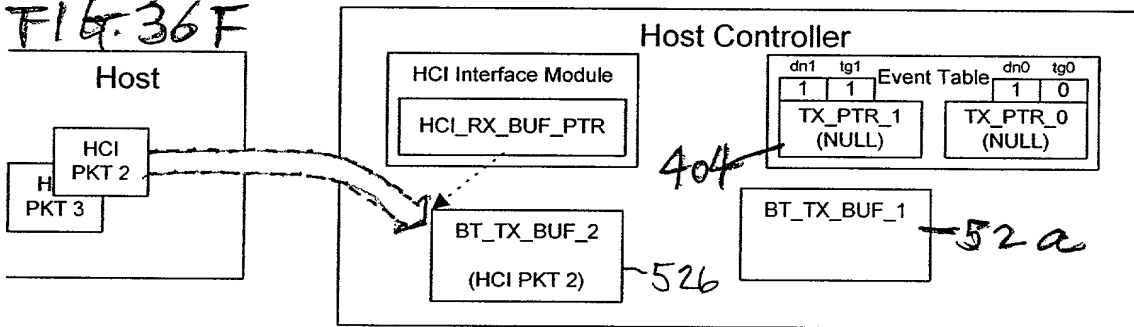
Step 5: While the Bluetooth Module is sending the HCI packet 1, the HCI Interface Module is receiving the HCI packet 2.

FIG. 36E



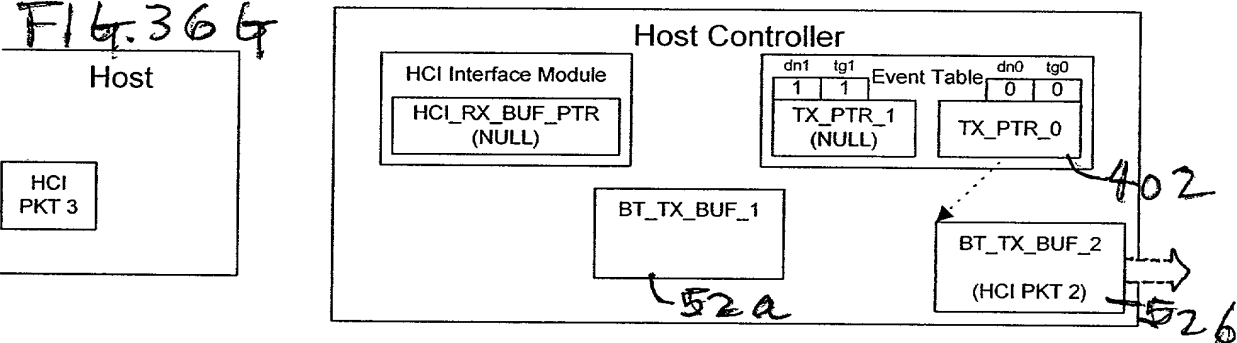
Step 6: Assuming that the HCI packet 1 is sent before the HCI packet 2 is received, the buffer 'BT_TX_BUF_1' is released.

FIG. 36F



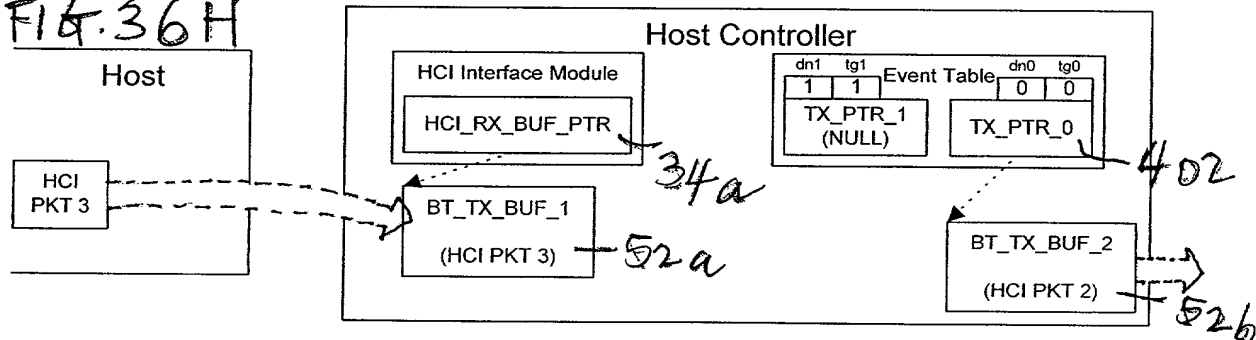
Step 7: After the HCI packet 2 has been received, the buffer 'BT_TX_BUF_2' is pointed by 'TX_PTR_0'.

FIG. 36G



Step 8: The free buffer 'BT_TX_BUF_1' is assigned to the HCI Interface Module again to receive another HCI packet.

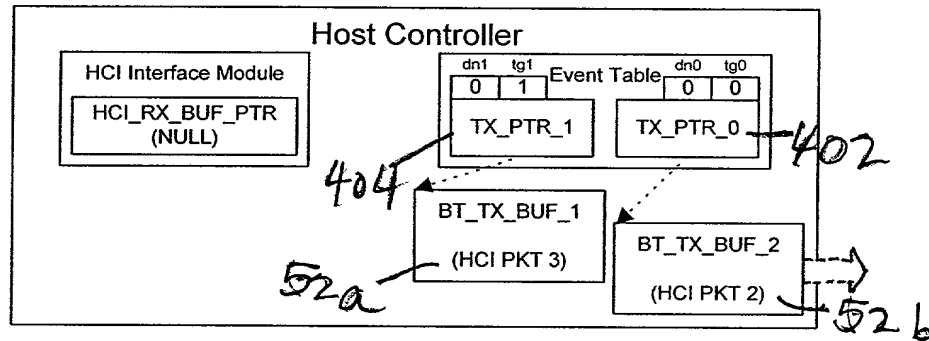
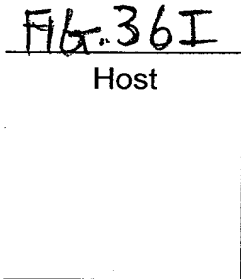
FIG. 36H



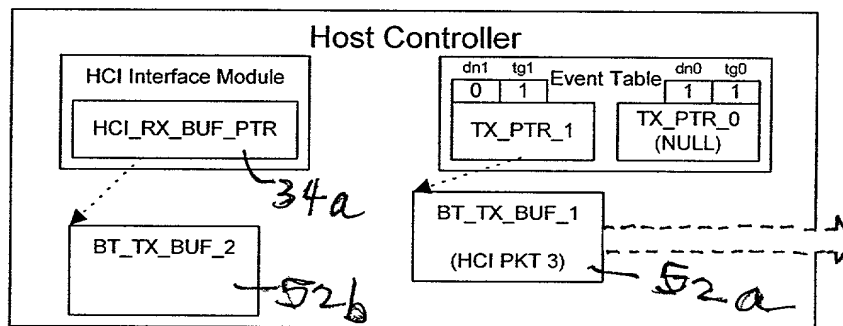
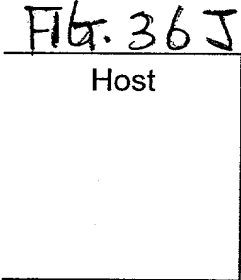
Example of Dual Pointer Buffer Scheme 1: TX Route

Page 3 of 3

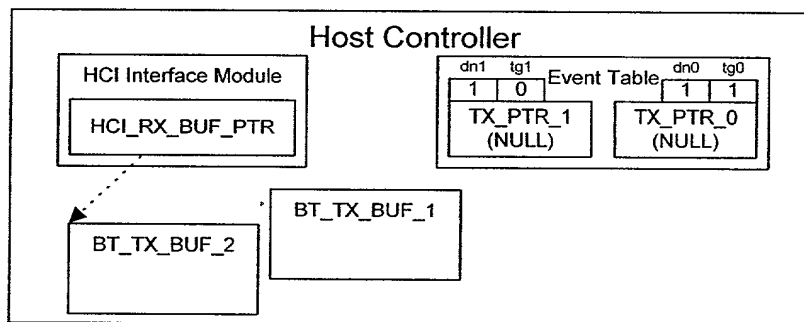
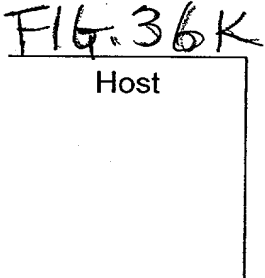
Step 9: Assuming that the HCI packet 3 is received before the HCI packet 2 is sent, 'TX_PTR_1' points to buffer 'BT_TX_BUF_1'.



Step 10: The free buffer 'BT_TX_BUF_2' is assigned to the HCI Interface Module again to receive another HCI packet.

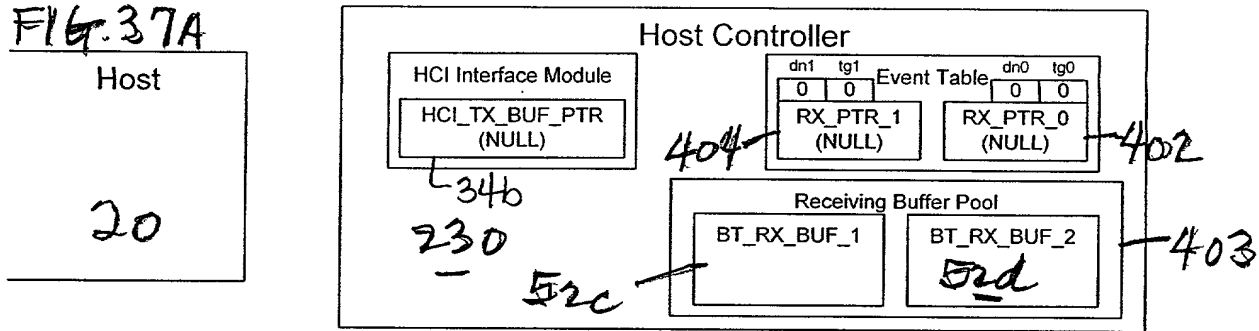


Step 11: After the HCI packet 3 has been transmitted, the buffer 'BT_TX_BUF_1' will be released.

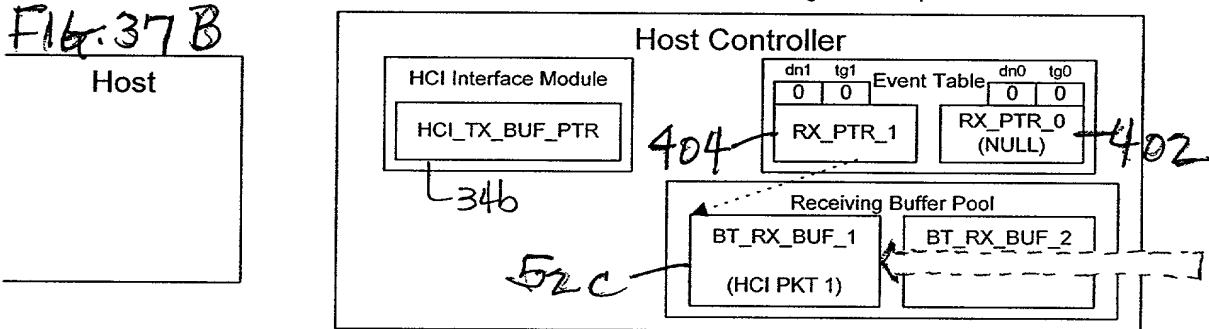


Now, the state of done and toggle is "1011". For the next transmission, 'TX_PTR_0' is selected to point the next outgoing packet.

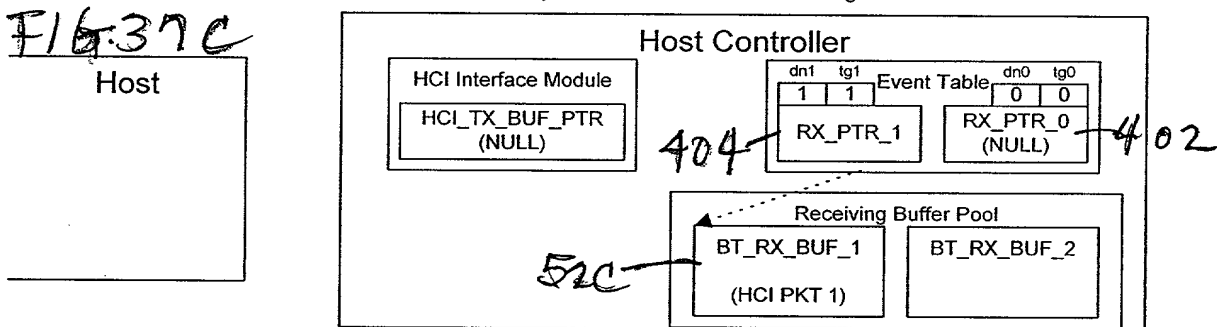
Step 1: After initialization, the value of pointers is 'NULL'. Assuming that two receiving buffer are available.



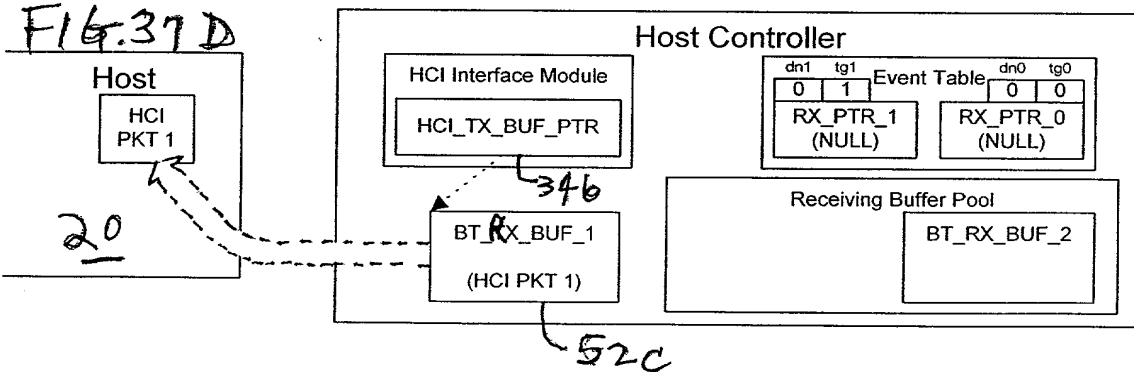
Step 2: The Bluetooth Module assigns buffer 'BT_RX_BUF_1' to receive the incoming Bluetooth packets.



Step 3: The buffer 'BT_TX_BUF_1' is released when any one of the three buffer releasing conditions is detected.

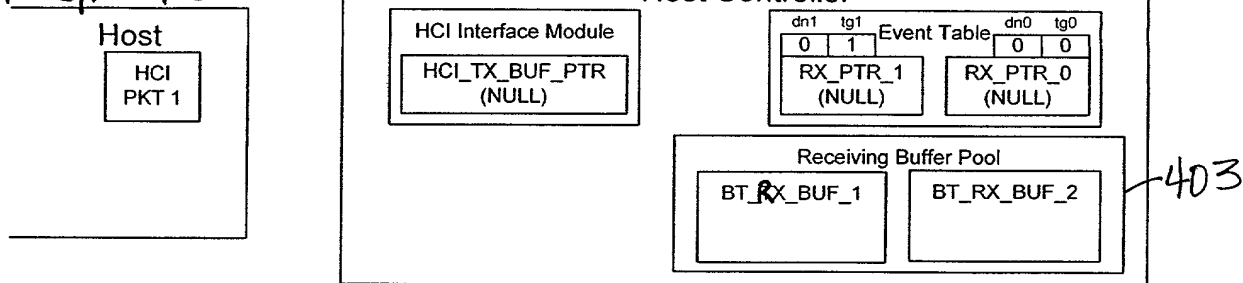


Step 4: Firmware releases this buffer 'BT_TX_BUF_1' and sends it to the HCI Interface Module. Then sets the done bit to 0.



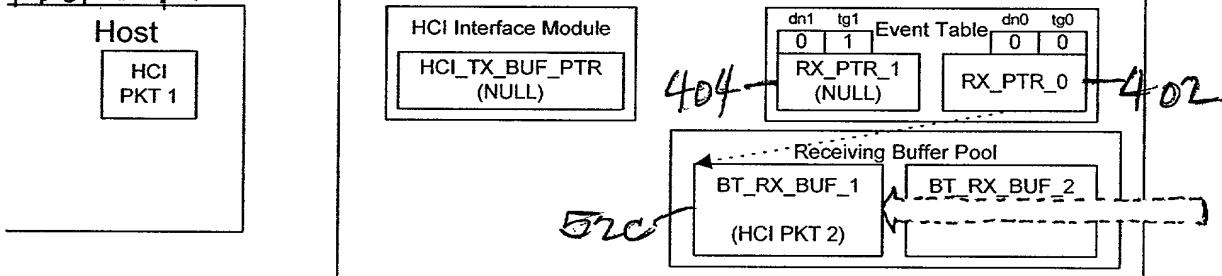
Step 5: After the HCI packet 1 is sent to the Host, buffer 'BT_RX_BUF_1' is released and put back to the receiving buffer pool

FIG. 37E



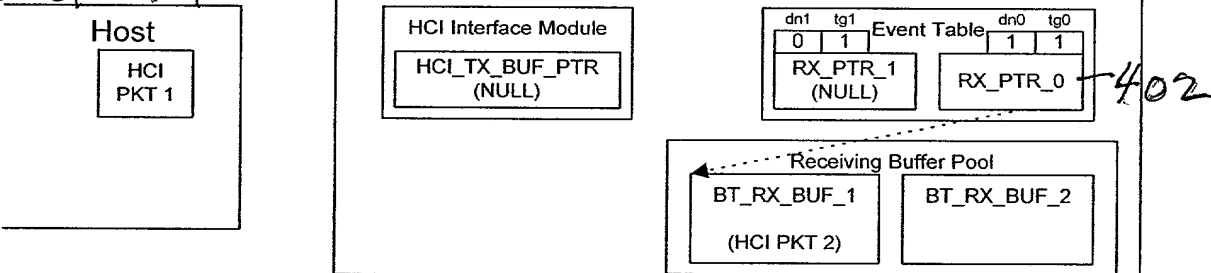
Step 6: The Bluetooth Module assigns buffer 'BT_RX_BUF_1' to receive the incoming Bluetooth packets.

FIG. 37F



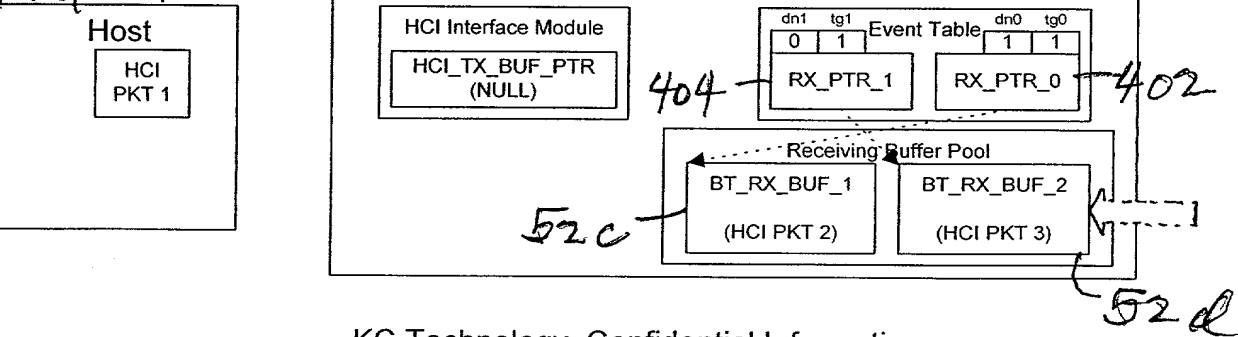
Step 7: The buffer 'BT_TX_BUF_1' is released when any one of the three buffer releasing conditions is detected.

FIG. 37G

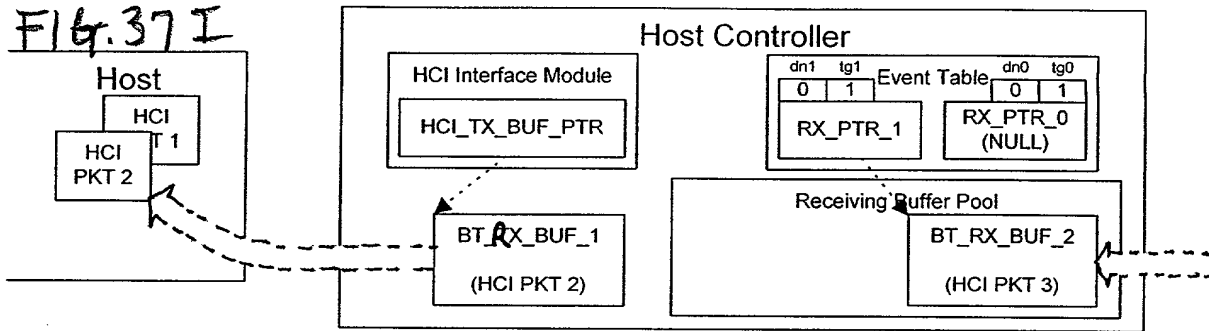


Step 8: Before buffer 'BT_RX_BUF_1' is removed by the firmware, another buffer is assigned to receive data.

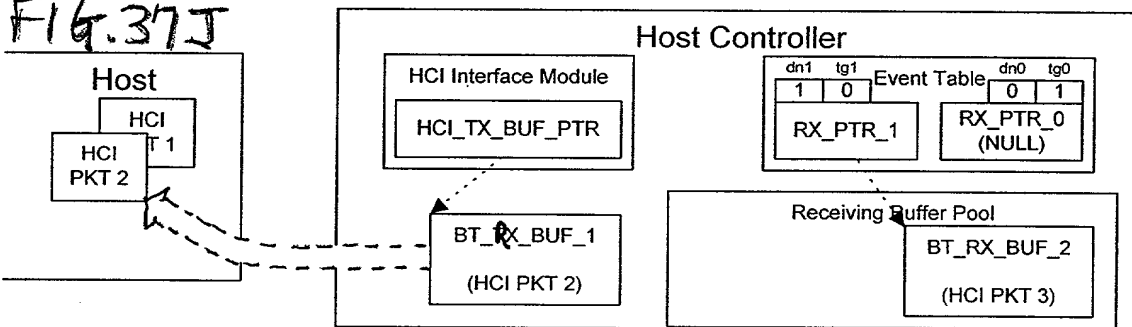
FIG. 37H



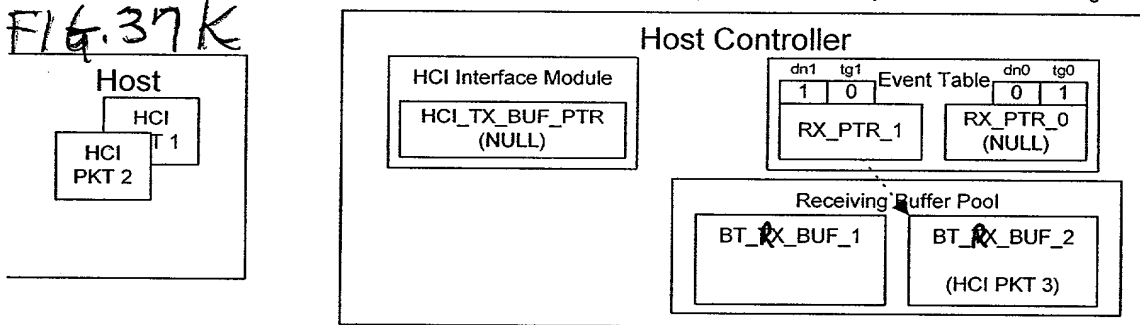
Step 9: Firmware releases this buffer 'BT_TX_BUF_1' and sends it to the HCI Interface Module. Then sets the done bit to 0.



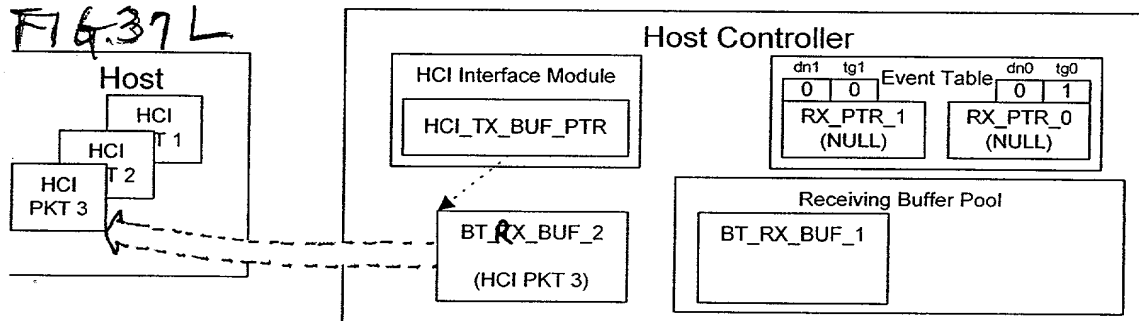
Step 10: The buffer 'BT_TX_BUF_2' is released when any one of the three buffer releasing conditions is detected.



Step 11: After the HCI packet 1 is sent to the Host, buffer 'BT_RX_BUF_1' is released and put back to the receiving buffer pool.



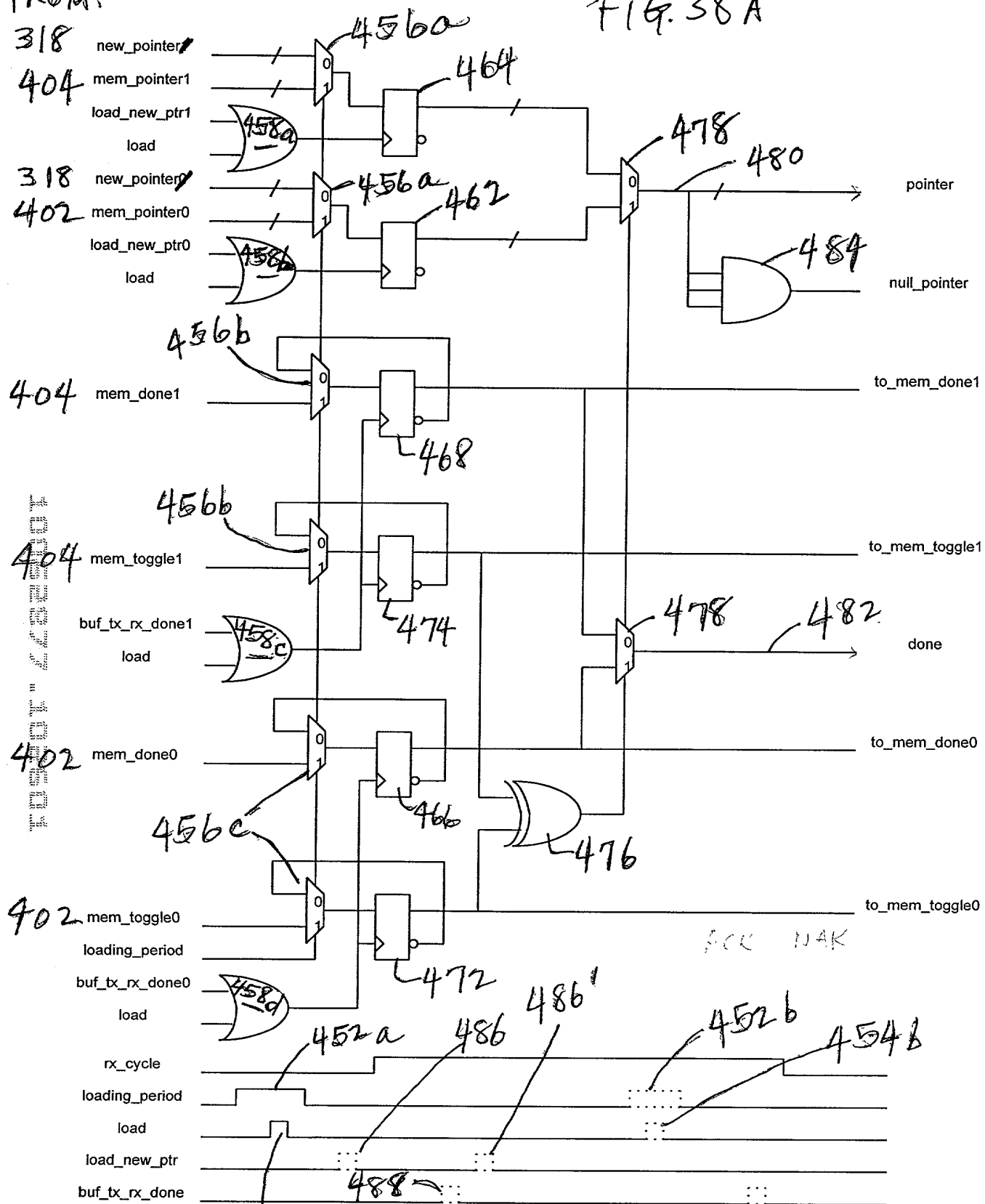
Step 12: Firmware releases this buffer 'BT_TX_BUF_1' and sends it to the HCI Interface Module. Then sets the done bit to 0.



Dual Pointer Buffer Scheme 1: Hardware implementation

FROM:

FIG. 38A



454a

FIG. 38B

KC Technology, Confidential Information

case 1: An interrupt which is generated by the Packet Controller of the BT module indicates that an incoming HCI packet is received

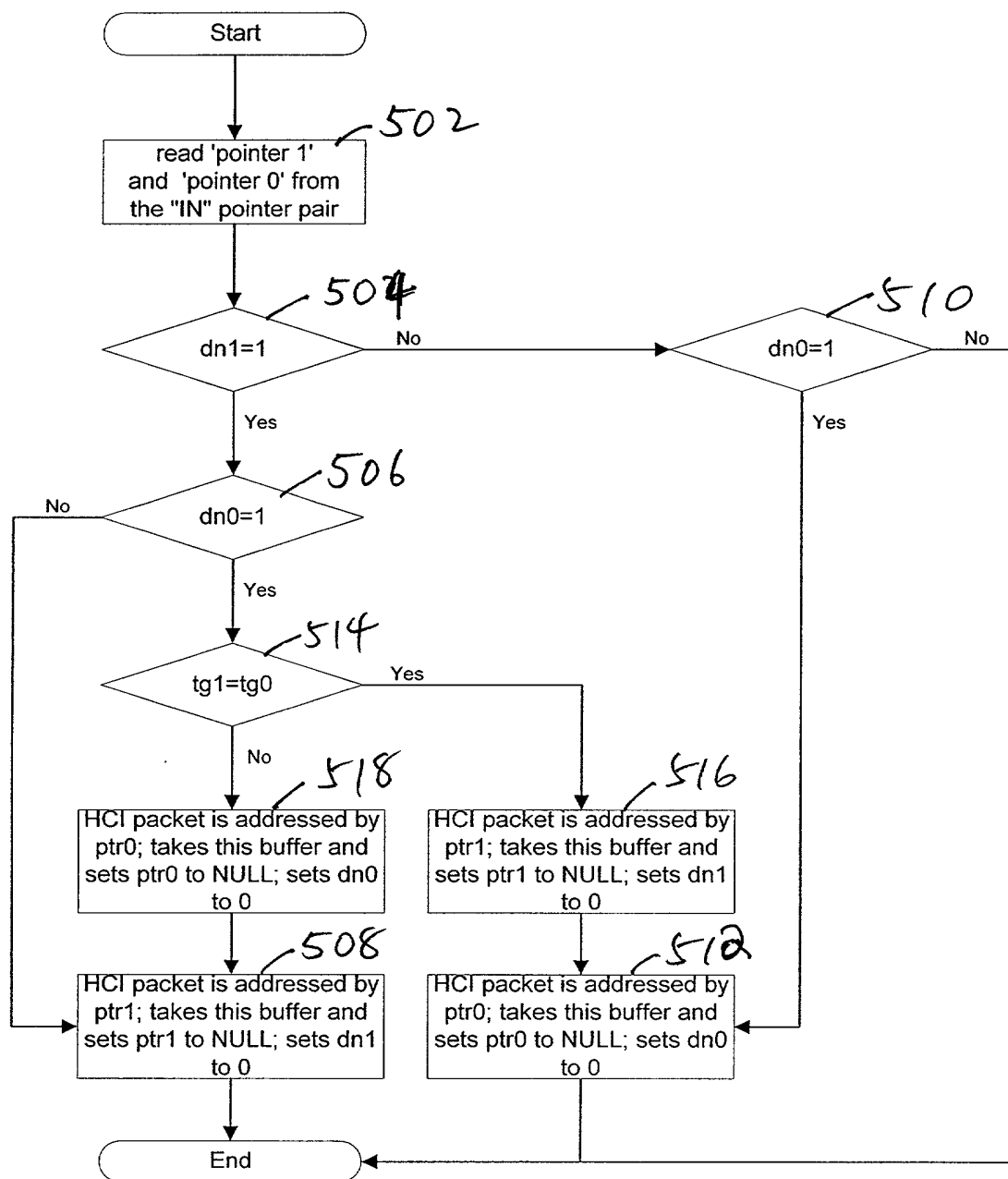


FIG. 39

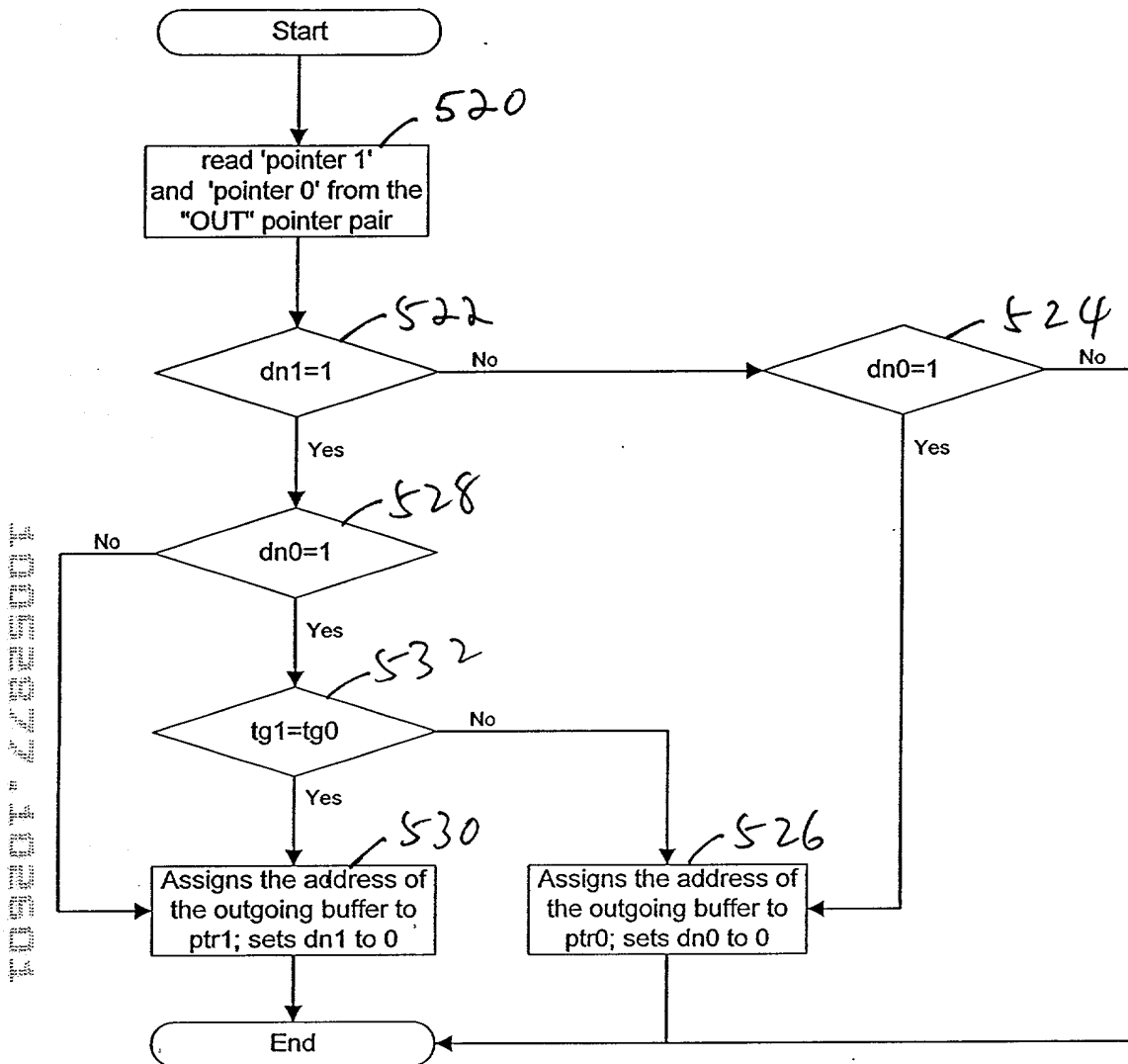


FIG. 40

case 2: An interrupt which is generated by the Packet Controller of the BT module indicates that an outgoing HCI packet is sent

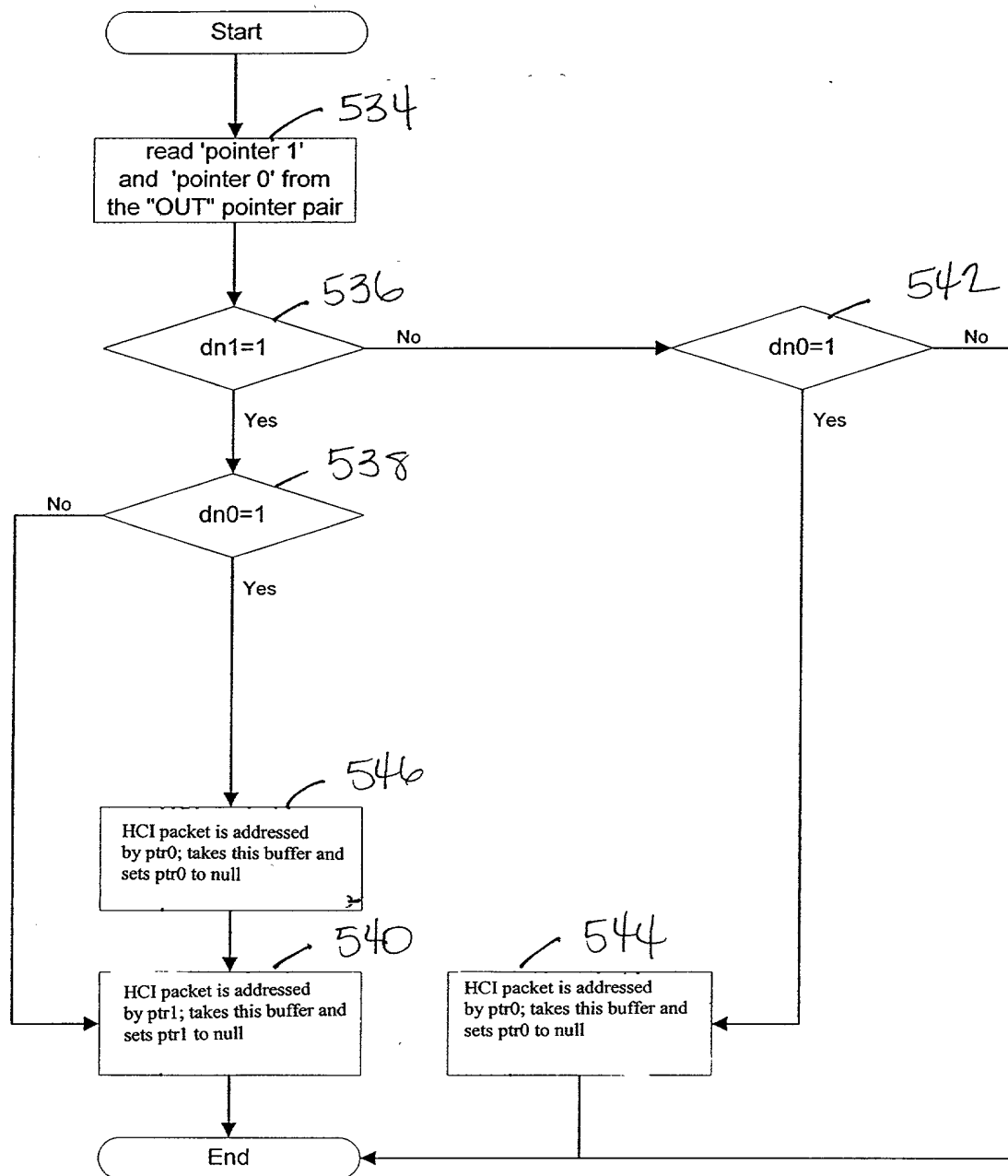


FIG. 4

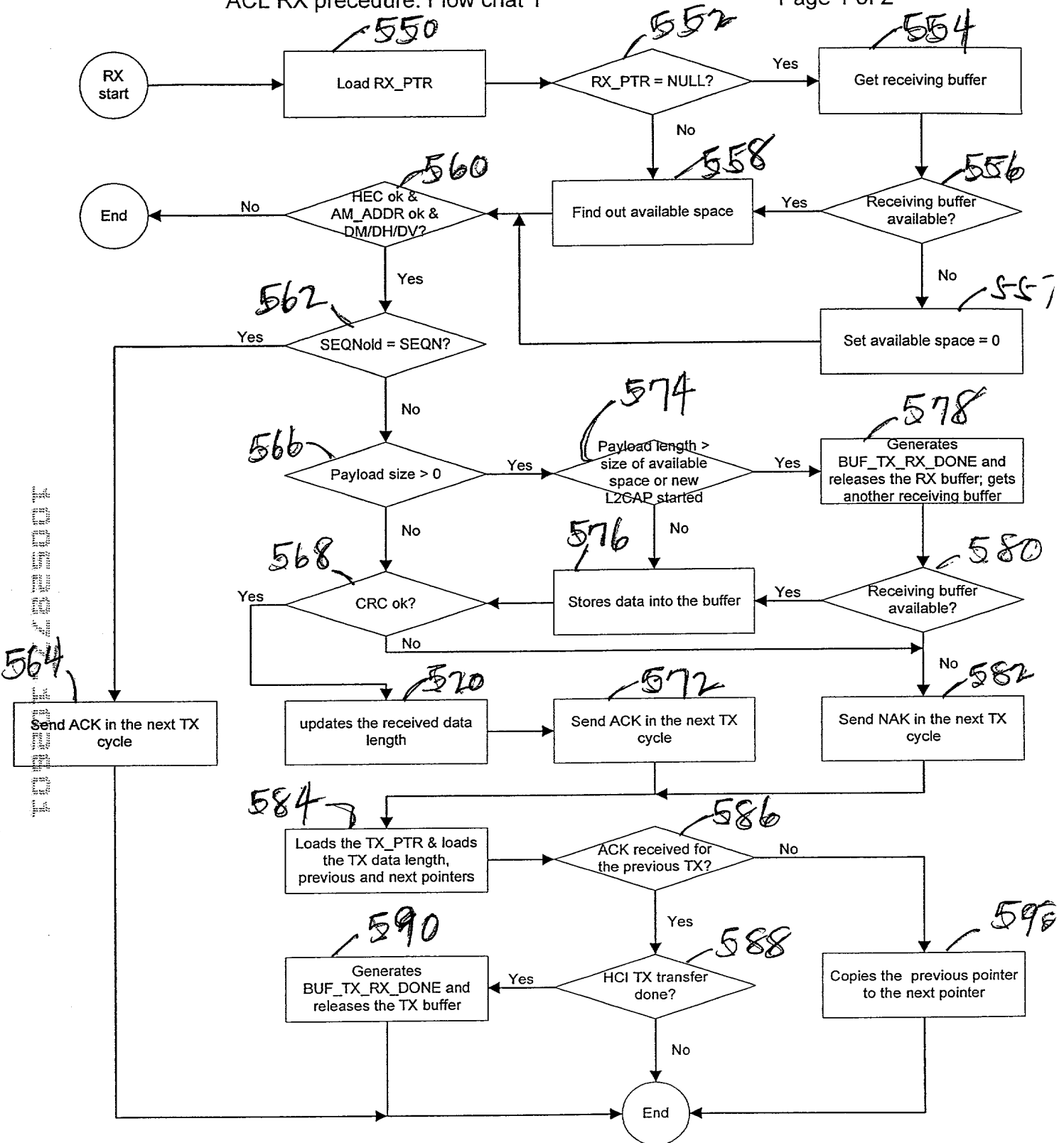
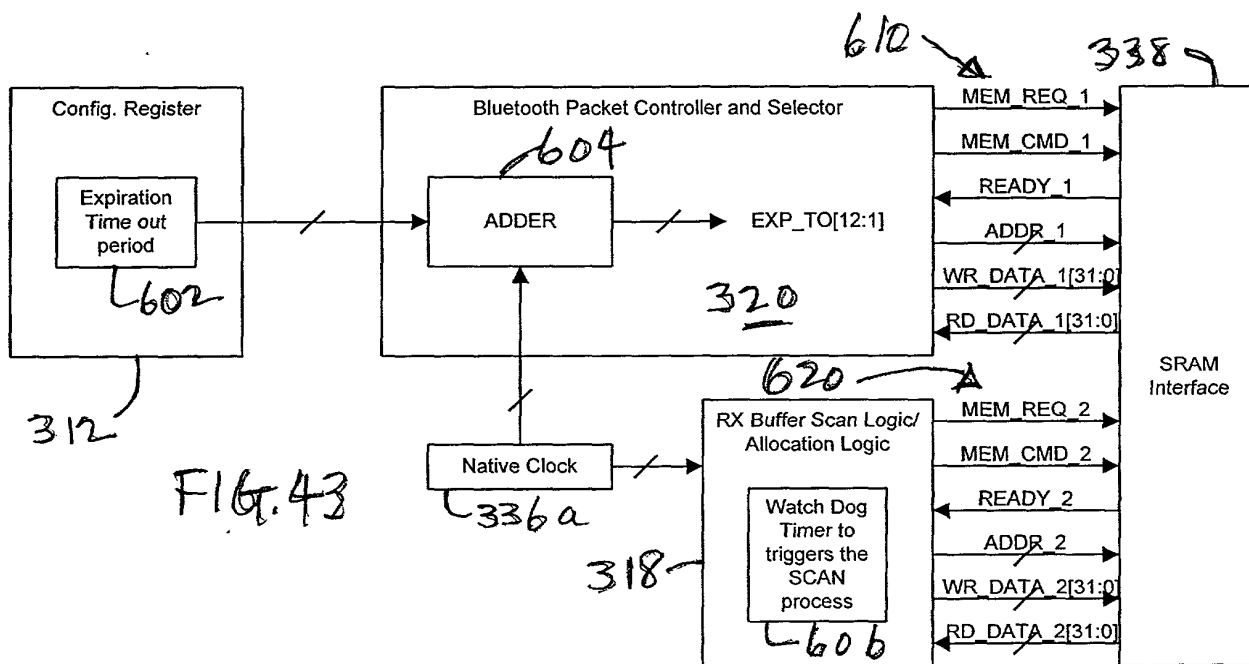


FIG. 42



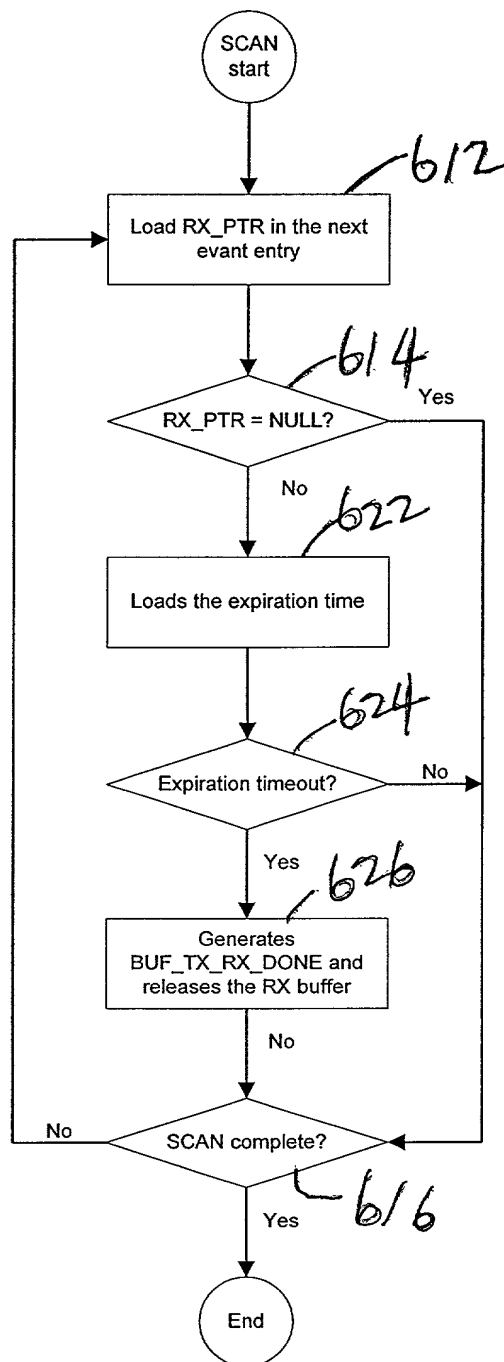


FIG. 44

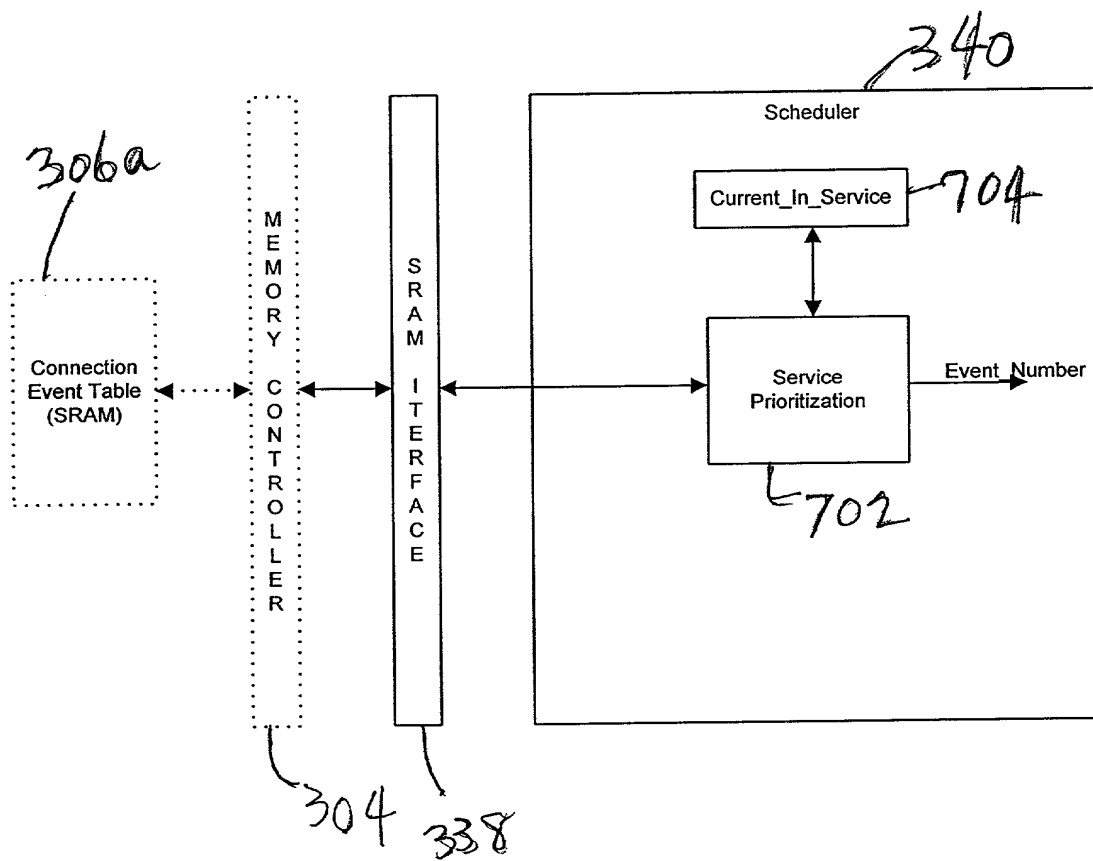


FIG. 45

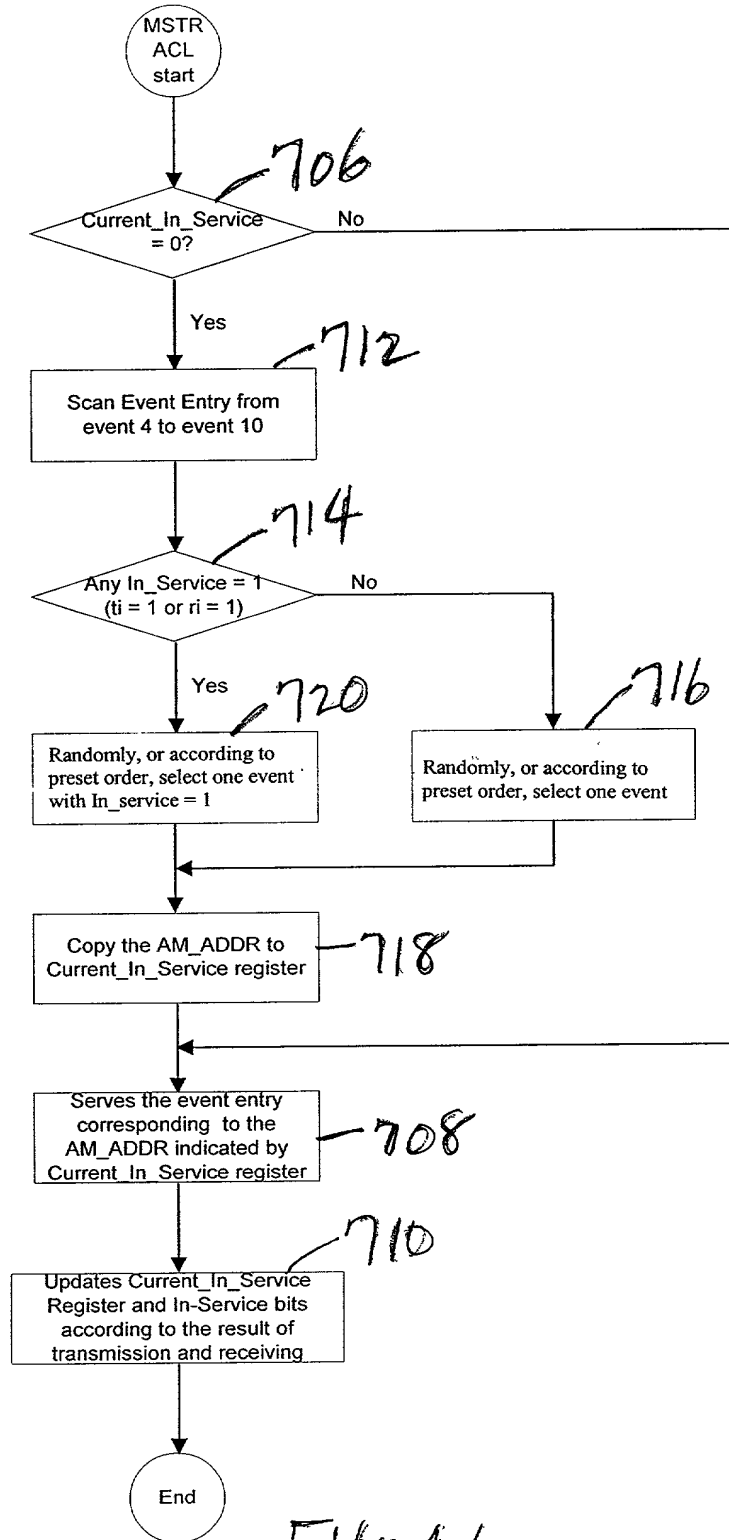


FIG. 46